MAYO 86 - 300 ptas.

TodoSpectrum

AÑO II - Número 21

REVISTA EXCLUSIVA PARA USUARIOS

PROBAMOS

Art Studio, el Spectrum hecho arte

PROGRAMAMOS

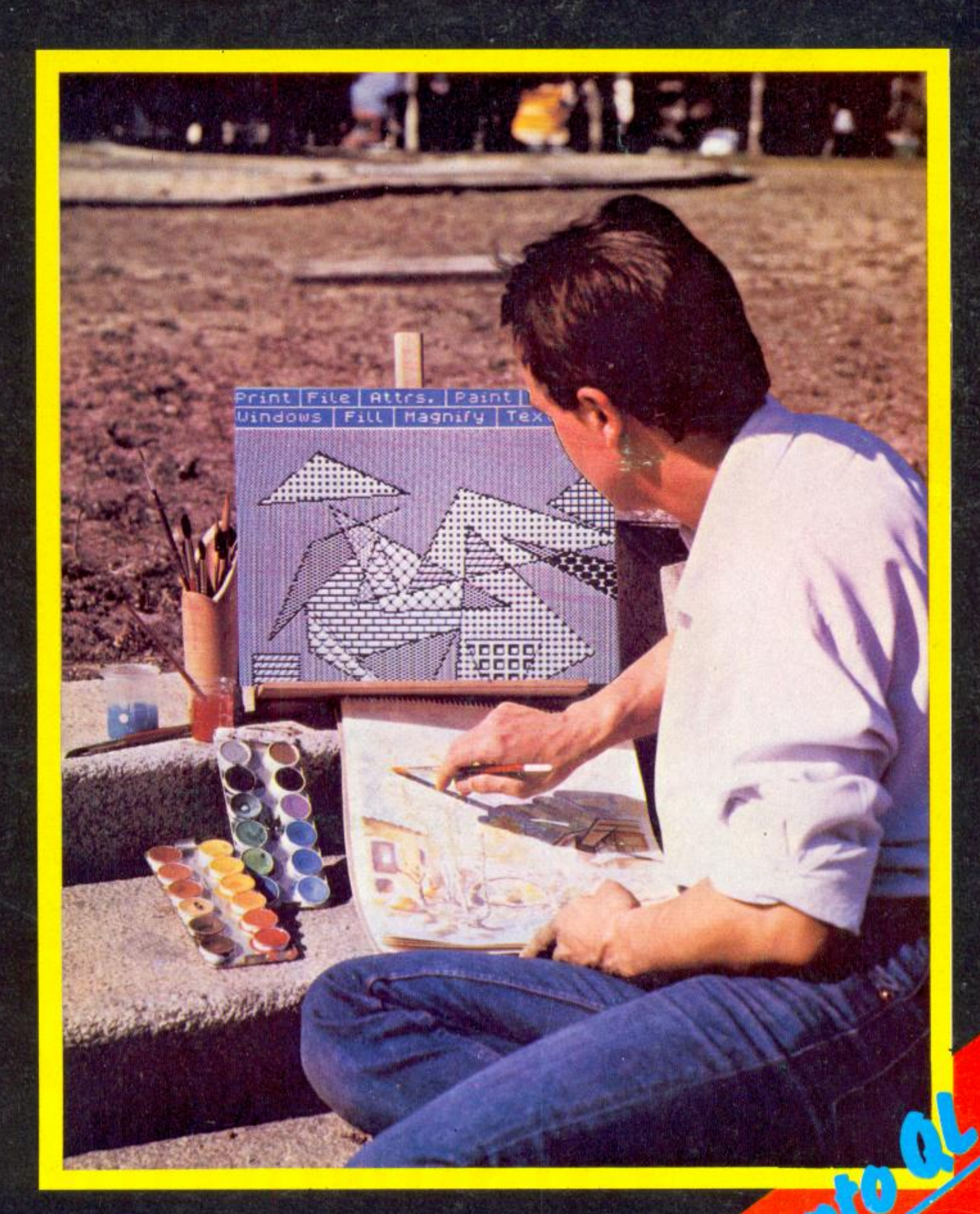
Compresor de pantallas

DESCUBRIMOS

Los bugs del sistema operativo

INVESTIGAMOS

Sistemas operativos multitarea



Suplemento QL

Quien es quien en el mundo del QL

SPECTRUM 128
EL SUMMUM

Spectrum, como líder, marca un nuevo hito en la historia de los ordenadores familiares.

El Spectrum 128.

Gran capacidad de memoria. Teclado y mensajes en castellano, teclado independiente para operaciones numéricas y de tratamiento de textos...

Sinclair e Investrónica han desarrollado una auténtica novedad. En ningún lugar del mundo, salvo en los Distribuidores Exclusivos de Investrónica, podrás encontrar el nuevo Spectrum 128.

Sé el primero en tener lo último.

SPECTRUM 128. NOVISSIMUS



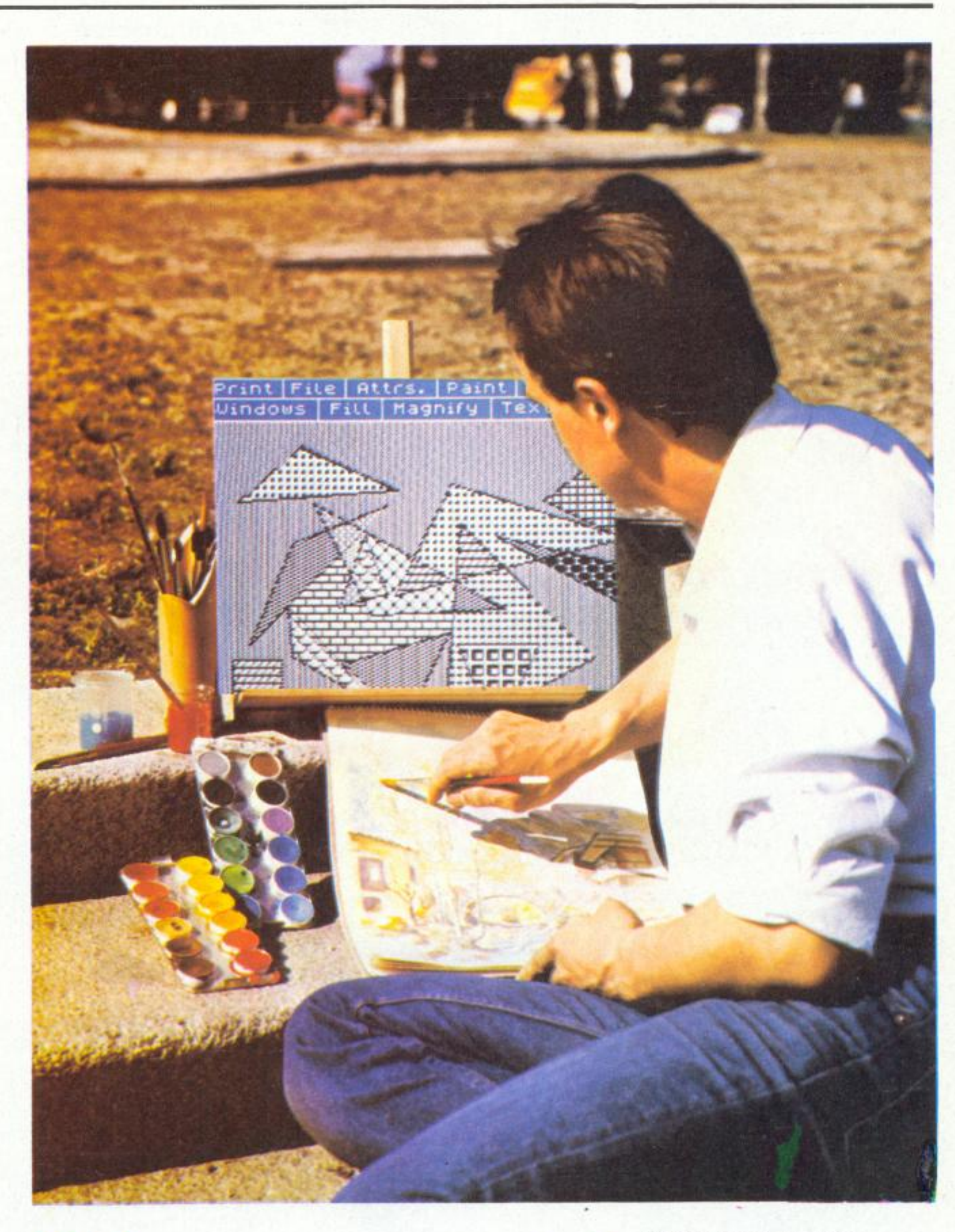
investronica

Tomás Bretón, 62.
Tel. (91) 467 82 10.
Telex 23399 IYCO E.
28045 Madrid

Camp, 80.
Tels. (93) 211 26 58 - 211 27 54.
08022 Barcelona

SUMARIO

AÑO II - Nº 21 - MAYO 1986



6 ART STUDIO, CUANDO EL SPECTRUM SE HACE ARTE

Análisis del más potente programa de dibujo desarrollado hasta el momento. Ventanas, iconos, menús descendentes y ratón trabajando en el Spectrum.

14 COMPRESOR DE PANTALLAS

Ponga a dieta sus pantallas: con esta rutina se comerán mucha menos memoria, pudiéndose almacenar simultánemente más de cinco en un Spectrum 48 K.

18 SWEEVO'S WORLD, GUIA DEL HACKER

Manuel Arana continúa desmenuzando programas. Le ha llegado el turno al último juego de Gargoyle Games, Sweedo's World. Descubrimos todos sus secretos.

24 NOTICIAS

Amstrad Consumer Electronics compra Sinclair Research... ¡y nosotros con estos pelos!

26 FACTORES PRIMOS

Juan Ignacio Perea, ganador del concurso matemático «El número más largo», comenta a fondo su programa.

31 SUPLEMENTO QL

¿Quién es quién en el mundo del QL?
Una exhaustiva relación de los fabricantess, distribuidores, software y hardware para el ordenador más polémico de Sinclair.

Los Bugs del Sistema Operativo
Ningún ordenador está libre de errores. El QL tampoco, pero si los conoce podrá evitarlos.

Sistemas Operativos Multitarea

40 APRENDIENDO LENGUAJE MÁQUINA

Terminamos el análisis del juego de instrucciones del Z-80 y ofrecemos una completa tabla con sus características.

46 PROGRAMAS

Química, física y juegos están representados en la sección de programas de este mes: Sistema periódico, Dinámica y Mastermind.

SERVICIO DE EJEMPLARES ATRASADOS



Complete su colección de

Todospectrum

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 2 - 300 ptas.
Gráficos profesionales/Desplazamiento pixel a pixel/Utilización de rutinas/Construcción del interface centronics/Programas de utilidad para microdrive/Rutina reset en código máquina/Análisis del editor de textos Tasword/Interfaces para impresoras/Programas.

Núm. 3 - 300 ptas.
Novedades sonimag'84/Ampliando el Basic/Programas para ordenar programas/Gráficos con el VU-3D/Lenguaje Forth/Archivos en microdrive/Programación de un interface de impresora/Programas.

Núm. 4 - 300 ptas.
De profesión: programador/Consola para el Spectrum/Comparación código máquina-Basic/Análisis programa contabilidad/Calendario/Pascal/Programas.

Núm. 5 - 300 ptas.
Floppys para Spectrum/Diseño asistido por ordenador/64 Caracteres por línea/Juego de la vida/Pascal/Así hacemos las portadas/Control de evaluaciones/Programas.

Núm. 6 - 300 ptas.
Representación de funciones/Todos los caminos conducen a la ROM/Juegos/Pascal/Construcción de un lápiz óptico/Programas de gestión. El SITI/Logo: torgugas para todos/ Interrupciones del Z-80/Programas.

Núm. 7 - 300 ptas.
Del 48 al PLUS paso a paso/¿Plotter para Spectrum?/Juegos/Libros de código máquina/Lápiz óptico. Programación del montaje/El LOGO en la escuela/Pascal/Floppys para Spectrum/Programas.

Núm 8 - 300 ptas.
Amplía tu memoria... a 48 K/Arquitectura: análisis del PREYME/Juegos/FORTH. Nociones básicas/Una clave, please/QL Magazine. Ultimas novedades, análisis de software, Lenguajes/Aula informática con Spectrum/Programas.

Núm 9 - 300 ptas.
Spectrum parlanchín/Juegos/Aula informática con Spectrum/Análisis: Comercial 4/Pascal/Periféricos: Wafadrive/QL Magazine: EASEL lo mejor de PSION. Música con QL/Desplazamiento Pixel a Pixel, aportación de lectores/Programas/Programer II.

Núm. 10 - 300 ptas.
Discos: invesdisc 200/Juegos/Dos programas simultáneos/Protección del software/Conozca extremadura, consulte a su ordenador/Desensamblador Z-80/Sofware educativo/QL Magazine: novedades Informat, Hoja de cálculo, Ajedrez/Construya su propio Joystick/Pascal/programas.

DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE ZX
(sin necesidad de encuadernación)

Núm. 11 - 300 ptas.
Actualidad/La otra cara del LOGO/Juegos/El Spectrum habla castellano/SOFTAID ayuda para Etiopía/S.O.S. aquí el Spectrum/Dibujar con lápiz óptico/QL Magazine: Procesador de textos. Teclas de función programables/Programas.

Núm. 12 - 300 ptas.
Actualidad/Inteligencia artificial/Lápiz óptico dk'TRONICS/Juegos/Análisis/Bingo/Z-80 PIO/Código máquina/Análisis: MASTERFILE/Programas.

Núm. 13 - 300 ptas.
Actualidad/Discos: Discovery 1/Juegos/Inteligencia artificial/Un nuevo sistema operativo/QL Magazine: Archive, Cartridge doctor. Aplicaciones comerciales/Código máquina/Programas.

Núm. 14 - 300 ptas.
Actualidad, Spectrum 128/Cálculo de estructuras para ingenieros y arquitectos/HELP utilidades en microdrive/Juegos/El microdrive ese desconocido/Código máquina/QL Magazine: GRAPHIC QL. Juegos. Discos de 720 K/Un nuevo operativo/Programas.

Núm. 15 - 300 ptas.
Actualidad/Spectrum 128/Un nuevo operativo/Círculos redondos/Juegos/Utilidades: BETA-BASIC/QL Magazine: Introducción al SUPER BASIC. Nuevas utilidades/Hardware: Puertas lógicas/Código máquina/Programas.

Núm. 16 - 300 ptas.
Actualidad/Cinco horas con SCREEN$/Hardware práctico/Cálculos de infinita precisión/Juegos/Un nuevo operativo/QL Magazine: Gráficos en SUPER-BASIC. Dibujando con ratón. Archivos con Archive. Programa/La última batalla, Juego estratégico.

Núm. 17 - 300 ptas.
Actualidad/Gráficos interactivos/Juegos/Código máquina/Un nuevo operativo/Trucos de programación/QL Magazine: Radiografia del QL. Gráficos en SUPER-BASIC/Libros/Programmas.

Núm. 18 - 300 ptas.
Actualidad/Introducción al C/Libros/Juegos/De cinta a microcinta/Visión panorámica de los microprocesadores más comunes/QL Magazine: Copy de grises. Microprocesadores 68000, una familia numerosa/Curioseando en la ROM/Programas.

Para hacer su pedido, rellene este cupón HOY MISMO y envíelo a:

Todospectrum Bravo Murillo, 377
Tel. 733 96 62 - 28020 MADRID

Ruego me envíen los siguientes ejemplares atrasados de TODOSPECTRUM .... al precio de 300 pts.

El importe lo abonaré
□ POR CHEQUE □ CONTRA REEMBOLSO □ CON MI TARJETA DE CREDITO □ AMERICAN EXPRESS □ VISA □ INTERBANK

Número de mi tarjeta: ______________________

Fecha de caducidad .... Firma

NOMBRE ....

DIRECCION ....

CIUDAD .... C. P. ....

PROVINCIA ....

DIRECTOR:
Enrique F. Larreta
REDACTOR JEFE:
Emiliano Juárez
REDACCION:
Ignacio Borrell, Octavio López,
Antonio del Rio
DISEÑO:
Ricardo Segura y Benito Gil

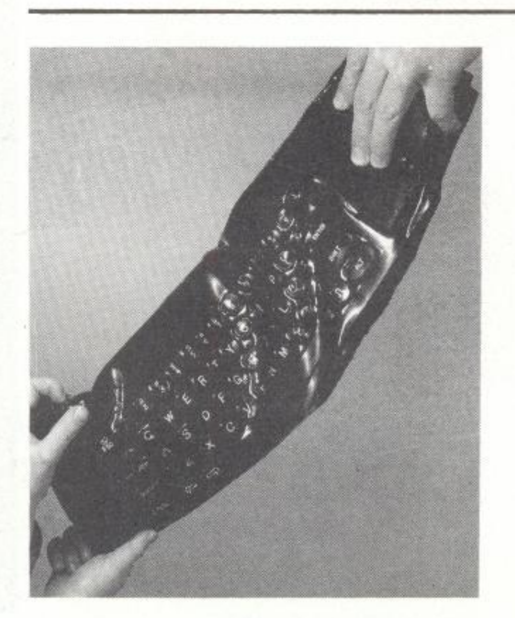
Editado por PUBLINFORMATICA, S. A.
Bravo Murillo, 377. 5.º A. Tel.:
733 74 13 - 28020 Madrid
Presidente:
Fernando Bolin
Director Editorial Revistas de Usuarios:
Juan Arencibia
Director de Ventas:
Antonio González
Producción: Miguel Onieva
Servicio al cliente:
Julia González. Tel.: 733 79 69
Administración:
PUBLINFORMATICA, S. A.
Publicidad Madrid:
Emilio Garcia
Dirección, Publicidad y Administración:
Bravo Murillo, 377. 5.º A. Tel. 733 74 13.
Télex: 48877 OPZX e 28020 Madrid
Publicidad Barcelona:
Lidia Cendrós. Pelayo, 12. Tels. (93)
318 02 89 - 301 47 00, ext. 27 y 28.
08001 Barcelona
Depósito legal: M-29041-1984
Distribuye S.G.E.L. Avda. Valdelaparra,
s/n. Alcobendas (Madrid).
Fotomecánica: Karmat, C/ Pantoja, 10.
Madrid.
Fotocomposición: Espacio y Punto
Imprime: Héroes, C/ Torrelara, 8. Madrid.
Distribuidor en VENEZUELA,
SIPAM, S. A.
AVD. REPUBLICA DOMINICANA, EDIF.
FELTREC - OFICINA 4B BOLEITA SUR
CARACAS (VENEZUELA)
Esta publicación es miembro de la
Asociación de Revistas de
Información **ari** asociada a la
Federación Internacional de Prensa
Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia
relacionada con suscripciones a:
TODOSPECTRUM EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5º
28002 MADRID
(Para todos los pagos reseñar solamente
TODOSPECTRUM)
Para la compra de ejemplares atrasados
diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377. 5.º A
Tel. 733 74 13 - 28020 MADRID

Si deseas colaborar en TODOSPECTRUM
remite tus artículos o programas a Bravo
Murillo, 377. 5.º A. 28020 Madrid. Los
programas deberán estar grabados en
cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada
colaboración aisladamente, estudiando su
complejidad y calidad.

EDITORIAL

¿ADIOS AL QL?



La compra de Sinclair Research, Ltd. por Amstrad Consumer Electronics cierra un largo año de especulaciones sobre el futuro económico de esta empresa. Cuando la situación parecía mejorar para Sinclair, que incluso presentó un nuevo ordenador —el Spectrum 128— y había pagado diez de los quince millones de libras que adeudaba, Sir Clive aceptó la oferta de Amstrad, cediendo los derechos de fabricación, distribución y venta de sus productos a su máximo rival.

De las primeras declaraciones de Alan Sugar, presidente de Amstrad se desprende que se potenciará el Spectrum, lanzándose en breve un modelo de 128 K con cassette incorporado y probablemente con un port de joystick.

El QL ha salido mucho peor parado del acuerdo entre ambas empresas. Alan Sugar afirmó: «No veo mucho futuro en él. En su forma actual ha muerto». El presidente de Amstrad, que debe ver más futuro en el Z-80 y el CP/M, dejó abierto un pequeño resquicio a la esperanza al asegurar que sus ingenieros analizarían a fondo la máquina y quizás la resucitasen añadiéndole una unidad de discos de tres pulgadas y suprimiendo los microdrives.

Entre tanto, Amstrad se ha comprometido a mantener durante siete años un stock de piezas que garantice la asistencia técnica y los repuestos.

Pero sea cual sea el futuro del QL, mientras sus usuarios manifiesten el mismo interés y dinamismo de siempre continuaremos apoyándoles desde estas páginas y ofreciéndoles nuestro habitual suplemento QL.

TODOSPECTRUM

ART STUDIO

CUANDO EL SPECTRUM SE HACE ARTE



Hasta que llegó el Macintosh de APPLE, las ventanas, iconos, ratones y menús concatenados (conocidos colectivamente como WIMP's) eran un lujo sólo al alcance de los compradores más solventes del mercado de ordenadores. Sin embargo, la popularización de este sencillo entorno de manejo ha hecho que la industria de software inunde el mercado con sistemas orientados hacia el WIMP para micros tan diversos como el IBM PC y el SPECTRUM.

RAINBIRD, una filial de BT, ha incluido en su repertorio un excelente paquete de gráficos tipo WIMP llamado *The Art Studio,* de *Oxford Computer Publishing.* En la publicidad de este programa se asegura que transforma el SPECTRUM en un Macintosh, lo que resulta algo exagerado pero no demasiado alejado de la realidad.

Hay una voluntad consciente en su presentación que sugiere que los diseñadores se sintieron obligados a emular el Macpaint de APPLE. Sea verdad o no, el Art Studio, por 3.300 ptas. (precio aproximado al cambio de 14.95 libras), está tan cerca del Macintosh como es posible en un SPECTRUM, y supone un paso adelante respecto de los otros paquetes de diseño.

Desgraciadamente, el Art Studio incorpora una bestia negra: el LENSLOK. Para aquellos de vosotros que aún no os hayáis enfrentado con este verdadero reto a la paciencia, LENSLOK es un nuevo sistema de protección anti-pirata. En lugar de colores, códigos y contreseñas, este ingenioso sistema se utiliza alineando debedamente una lente especial que permite al usuario leer un número y así introducir correctamente el programa. A menudo son necesarios varios intentos para conseguirlo.

La recompensa por superar con éxito el LENSLOK es una pantalla blanca con un menú en forma de barra de color azul, dispuesto en la parte superior y un cursor en forma de flecha en el



centro. Hay tres métodos de manipular el cursor: las cuatro teclas de flechas en el teclado del SPECTRUM, el joystick o el ratón AMX (de Advanced Memory Systems).

Es preferible utilizar el ratón, porque, al contrario de los otros dos, permite dibujar las curvas con exactitud, describiendo un simple giro con la mano. En cambio, las curvas dibujadas con teclas o joystick tienden a ser irregulares y toscas.

DISEÑO

La característica más deliciosa del Art Studio es el uso de menús concatenados, un fenómeno relativamente nuevo en el SPECTRUM. Un menú de este tipo es una lista de funciones contenidas bajo una misma cabecera. Por ejemplo, la barra del menú en la parte superior de la pantalla que aparece en la Art Studio contiene sólo 11 opciones. Sin embargo, al seleccionar una de ellas con el cursor, aparece en pantalla una ventana llena de opciones.

Velocidad y flexibilidad son las dos mejores virtudes de los WIMP's. Además, para aquellos que sufren de «teclado-fobia», la posibilidad de seleccionar las opciones sin tener que pulsar teclas es una bendición. Esta sencillez de manejo se mantiene a lo largo del programa. Ilustraremos esto de una forma

más clara examinando los contenidos de los menús concatenados.

MENUS

Cuando la mayoría de la gente se encuentra con un paquete de diseño, lo que quiere hacer es dibujar cosas. La primera puerta a la que llamar es, por tanto, el tema de dibujo, ya que contiene los útiles básicos para dibujar. Componen este menú, un lápiz, un spray y una brocha, a la vez que un editor de brochas y una orden llamada INVERSE.

Si seleccionamos la opción lápiz, se accede a otro submenú que contiene un muestrario de 16 yipos de lápices. Algunos son para el trabajo con líneas finas, mientras que otros son más adecuados para sombrear o para dar ciertos efectos especiales. Cuando el comando INVERSE se encuentra en posición ON, el lápiz se transforma en una goma de borrar, con una posibilidad de selección de tamaños tan amplia com las opciones de los lápices.

El bote de spray es similar al lápiz, excepto que sus 8 opciones se refieren más a la densidad que a la forma. Cuando se usa, este spray reproduce fiélmente el efecto de los verdaderos, incluso hasta el extremo de que si se mantiene el spray pulsado durante mucho tiempo la densidad del trazo aumenta. Se puede utilizar esta opción en conjunción con el ratón para perfeccionar el estilo, antes de intentar hacer cuadros para el comedor. En tercer lugar está la brocha. De nuevo encontramos 16 opciones entre las que elegir, aunque escogiendo el editor de brochas, es posible cambiar una o todas ellas para acoplarlas a las necesidades de cada momento.

Para hacer más fácil la vida del artista, el Art Studio tiene un menú de Ampliación, que contiene funciones orientadas a ampliar una parte de la pantalla dos, cuatro y hasta ocho veces su tamaño normal. Esta es una gran ventaja para los detallistas, ya que permite realizar cambios pixel a pixel.

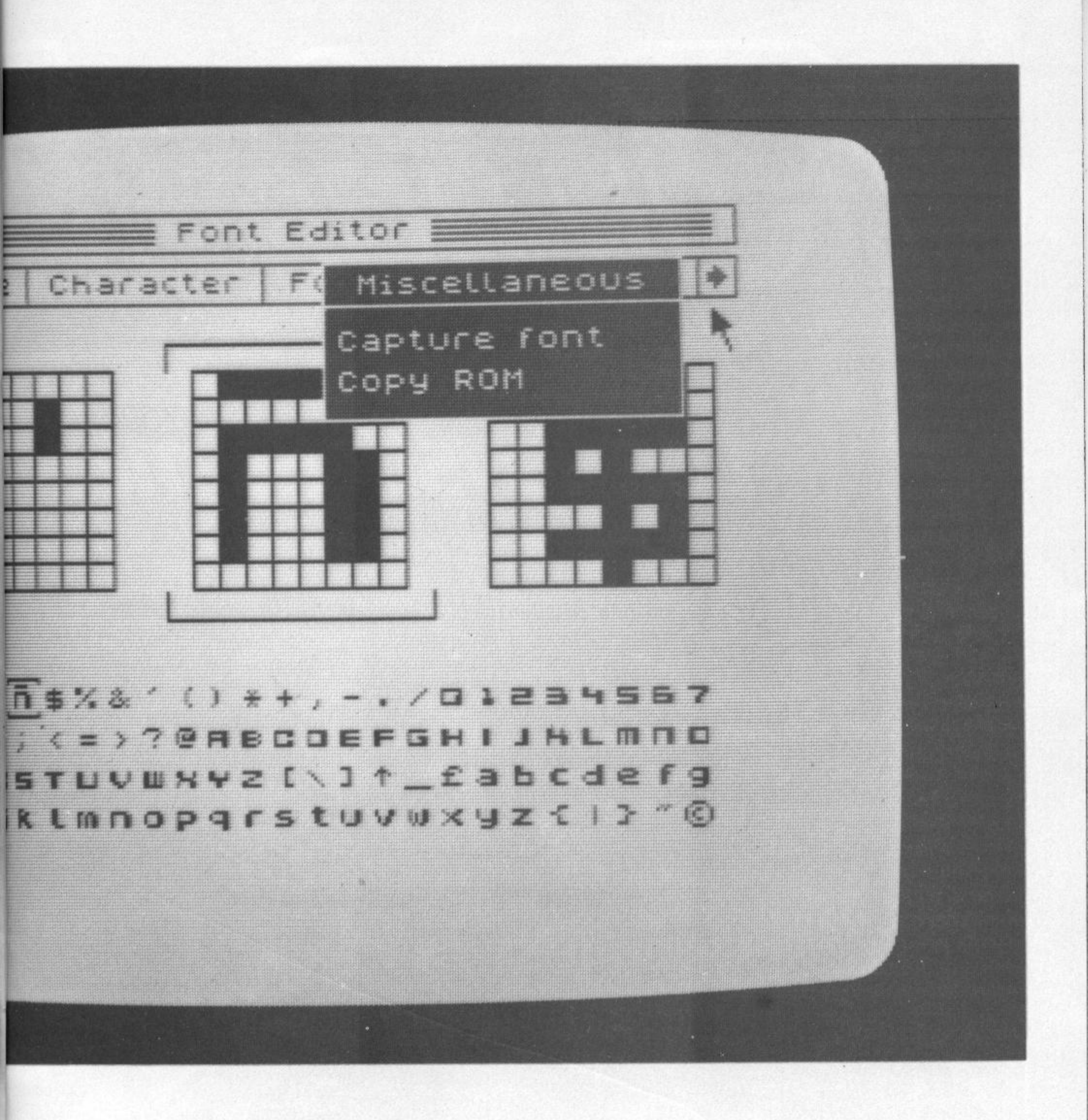
Las formas y estructuras son las principales preocupaciones del dibujante, la precisión y la claridad lo son del diseñador técnico. Para acomodarse a tales necesidades el Art Studio tiene un menú de formas que contiene utilidades que permiten qué determinadas formas geométricas sean dibujadas simple y rápidamente.

Círculos, rectángulos, triángulos y rayos pueden ser trazados sin necesidad de tener que recurrir a las teclas del SPECTRUM. Los círculos son especialmente fáciles de dibujar, usando una técnica muy similar a la utilizada en otros sistemas más costosos; primero, el centro se define presionando el ratón (tecla o joystick), y después la circunferencia se crea escogiendo un punto de ésta. El menú de formas permite ampliar y reducir las figuras hasta el tamaño deseado. Esto es especialmente útil para experimentar con los tamaños de los círculos. Para formas con lados rectos, hay dos rápida funciones que aseguran perfectas líneas verticales y horizontales.

Cuando los programadores se decidieron a crear un paquete de diseño gráfico que se pareciera al Macpaint, les interesó especialmente emular sus opciones de coloreado y rellenado de formas cerradas. Por tanto, no es sorprendente que dichas opciones del Art Studio, sean casi idénticas a las del Macpaint. Para asimilar lo que en el APPLE se llaman texturas o calidades sin mezclarlo con otros tipos de sombreado, el Art Studio tiene 3 formas de rellenado y un editor texturas.

Por supuesto, todas las funciones de rellenado se encuentran en un menú de llenado bastante completo. Seleccionando el rellenado sólido o liso, el primero del menú, se transforma la flecha del cursor en un rodillo de pintura, el cual, cuando es colocado en una figura cerrada, la rellena con el color de tinta que se esté usando, de forma muy rápida.

Si todo te parece familiar en el rellenado de color liso, espera a entrar en el relleno a base de entramados. Escondidas bajo esta opción se encuentran 32 diferentes tramas al estilo del Macpaint (64 si también contamos las in-



versas), cada una de estas puede ser aplicada a cualquier superficie en la pantalla. La aplicación de tales entramados se realiza en dos fases: primero, se llena la superficie con el color de la tinta y luego el entramado se estampa sobre todos los pixel afectados.

Aunque pudiera parecer lento, el uso de tramas es extremadamente rápido, al igual que la opción deshacer (undo), que permite que el último comando ejecutado pueda ser modificado. Si por ejemplo, se ha sombreado una superficie y no quedamos muy satisfechos con el resultado, este comando nos lleva de nuevo al estado original. Hay también una textura de cambio de color (wash) en el menú de rellenado que realmente no es tal rellenado en el más estricto sentido de la palabra. En vez de colorear o sombrear áreas o superficies vacías, esta opción permite aplicar un modelo sólo a determinados pixels a los que se les ha cambiado el color de tinta. Hay muchas aplicaciones para la opción «wash», pero una de las más efectivas, es el sombreado de un texto. Se puede remarcar una parte



Los Joysticks más

QUICKSHOT IV (3 en 1) Con mando para deporte

QUICKSHOT IV (3 en 1) Con mando de carreras

QUICKSHOT I MSX

QUICKSHOT I

QUICKSHOT VII - Portátil

QUICKSHOT IX Preciso y sensible

Los QUICKSHOT comercializados por SVI-España, S. A. son los únicos que tienen la GARANTIA OFICIAL SVI.

C&M

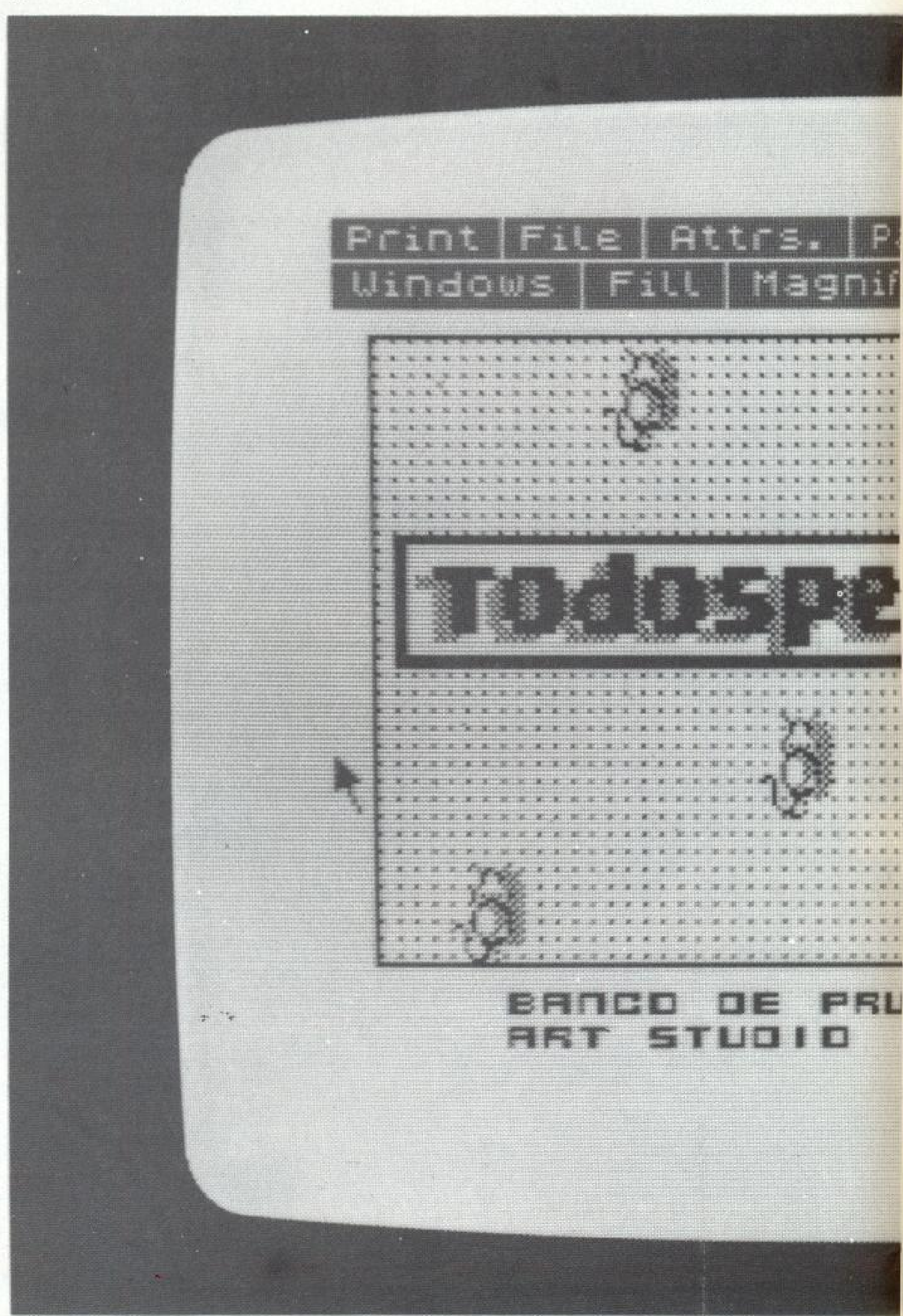
de un texto tiñiéndolo con cualquiera de las diferentes tramas.

El sombreado en blanco y negro puede producir interesantes efectos de penumbra si se usa correctamente. Por otro lado, para diseños más vistosos, se pueden cambiar los colores de acuerdo con un menú de atributos. Aquí es donde encontramos todos los colores normales del Spectrum, accediendo a ellos de forma natural desde el teclado.

Al igual que la mayoría de los paquetes gráficos para el SPECTRUM el Art Studio está afectado por las pobres posibilidades de manejo del color en el aparato. Aparte de presentar un aspecto pálido generalmente, los colores «chocan» entre sí, produciendo feos e involuntarios cuadrados en la pantalla. La causa de esto es la incapacidad del ordenador para manejar más de dos colores (uno para el papel y otro para la tinta) por caracter. Pueden resultar remedios útiles, reducir el dibujo a dos colores o usar una gran extensión del mismo color, haciendo que sea imposible que dos colores de papel o tinta entren en contacto. La primera vez que se usa el Macpaint, resulta asombrosa la forma en que los dibujos pueden quedar enmarcados y luego ser colocados en otra parte de la pantalla. Para muchos esto sólo es posible gracias a la presentación del mapa de bits del Macintosh. Por esto, nuestra sorpresa fue mayúscula cuando vimos el Art Studio realizando la misma tarea. La capacidad para realizar tales operaciones constituye uno de los rasgos más notables del Art Studio. Pero esto es sólo una parte de lo que se puede conseguir dentro del menú de ventanas.

Las funciones de «cortar y pegar» y «cortar, eliminar y pegar» son similares pero con útiles diferencias. La primera copia el contenido de una ventana y luego lo coloca donde queramos, dejando intacto el modelo original, mientras que la segunda, lo borra. Se pueden hacer varias copias de una ventana, escogiendo la opción múltiple del citado menú de ventanas.

Las ventanas son flexibles hasta el punto de cambiar de tamaño. Todas las ventanas, sin importar como sean de grandes o pequeñas, pueden ser modificadas de tamaño. Pueden ser aumentadas, reducidas, ensanchadas o estrechadas. La modificación tiene lugar en dos fases, al igual que la opción de entramado. Primero, la ventana se modifica en dirección vertical en un buffer interno, y después en dirección horizontal en pantalla. Esto va seguido de la ejecución en pantalla del primer paso. Cuando se reduce una ventana se pierden los detalles pequeños y el mencionado choque de colores se acentúa aún más. Por lo tanto, se consiguen los mejores resultados cuando se reduce una ventana que contiene sólo dos colores. Por último, indicar que las ventanas también pueden ser

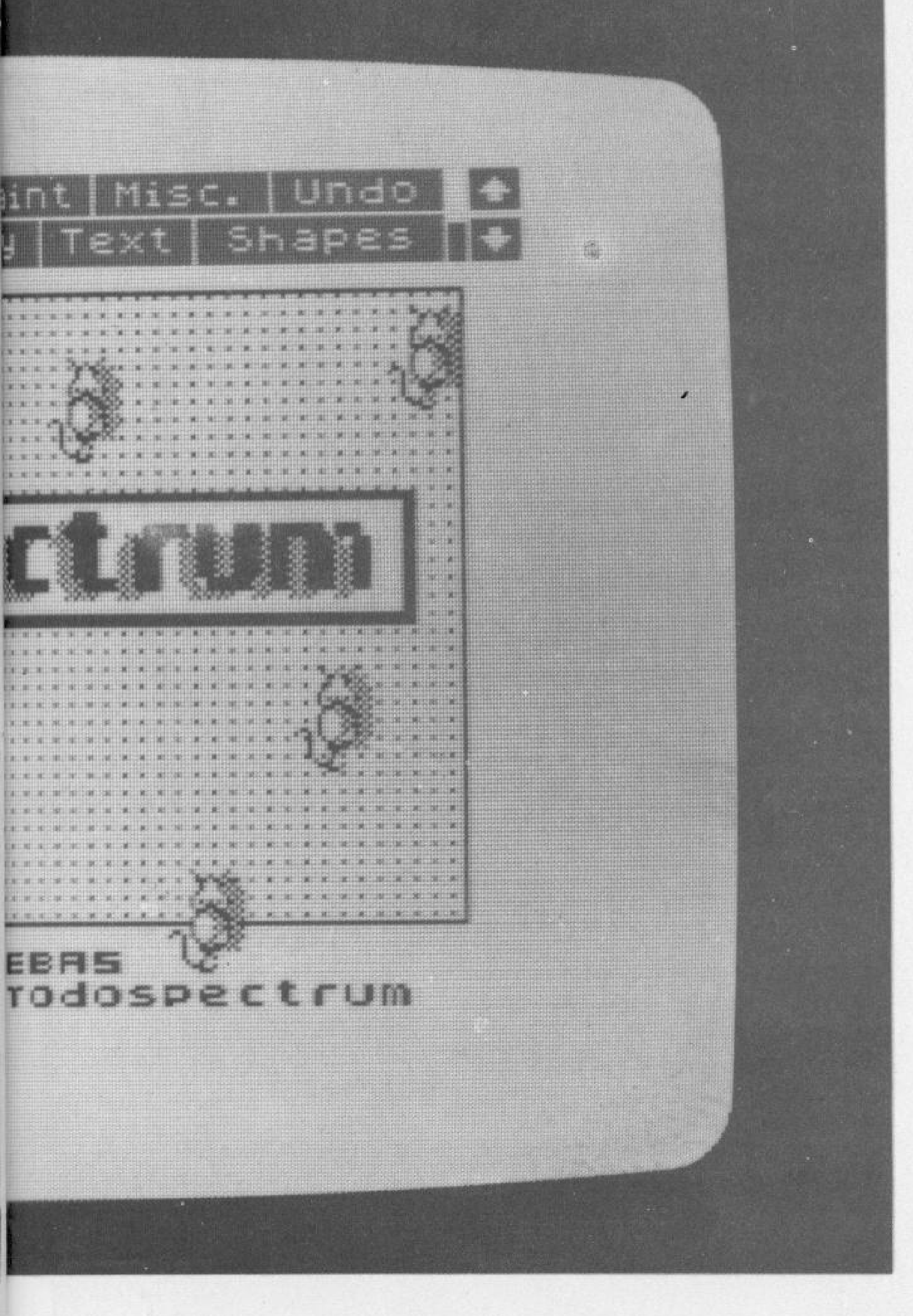


reproducidas a la inversa (como si se reflejaran en un espejo), duplicarlas o rotarlas.

DOCUMENTACION

Un programa tan complicado como el Art Studio requiere un complejo manual que guíe al usuario a través de los diferentes menús. RAINBIRD no se ha distinguido especialmente en este punto, produciendo un manual tan profesional como el programa al que acompaña. El folleto, de 58 páginas, está bien escrito y aclarado desde el principio hasta el final, concluyendo con un simple, pero no por ello menos útil, ejemplo de cómo hacer un dibujo. También se comenta en el manual un programa que permite a los compradores hacer una versión personal del Art Studio, copiando el programa en una cinta virgen.

Se personaliza la cinta en el sentido de que contiene parámetros específicos para determinados interfaces e impresoras. El Art Studio puede soportar diecisiete interfaces Centronics y RS232, así es difícil que alguien no pueda hacer una copia en papel, salvo que, por supuesto no tenga impresora.



PRECIOS

En principio el Art Studio es un programa en cassette que cuesta 14,95 libras, en Inglaterra, pero RAINBIRD también tiene una versión para los usuarios del microdrive de Sinclair y del Interface para discos de Kempston. No hay cambios en ellas, tan sólo, la adición de un programa que comprime las pantallas, una opción de arcos en el menú de formas, varios comandos de manejo de archivos y dos programas de instalación para microdrive y disco. El Art Studio no será capaz de hacer uso del compresor de pantallas, debido a la gran cantidad de memoria que esto requiere.

Sin embargo, RAINBIRD venderá la versión ampliada al precio reducido de 12 libras a los usuarios de A.S. que habiéndolo comprado ya, decidan elevar su equipo al microdrive o al disco.

CONCLUSION

El Art Studio es actualmente, uno de los más complejos y potentes paquetes de diseño que se pueden encontrar hoy para el SPECTRUM. Su mejor y más provechosa utilización proviene del diseño de pantallas de presentación para juegos comerciales. Aunque no se pueden reproducir las desbordantes imágenes de Dali, o la encendida pasión de Van Gogh, es posible producir pantallas de cabecera a nivel profesional en la medida de tu propio talento artístico.

Joaquín Mateos

COMPRESOR DE PA

Quien haya trabajado con gráficos en el Spectrum y haya tenido la necesidad de mantener varias pantallas en memoria a un mismo tiempo, se habrá dado cuenta de la cantidad de *bytes* que se «comen». Casi 7K por pantalla, que hacen que, aún disponiendo de un 48K, tras almacenar unas cuantas, nos quedemos sin sitio para otra cosa.

Las siguientes rutinas intentan paliar este problema, comprimiendo las pantallas de forma que ocupen menos memoria. Para ello aprovechan dos características que suelen aparecer en todas ellas:

La primera es que, por norma general, en toda pantalla existen zonas más o menos extensas en las que sólo hay tinta, sólo hay papel o se repite un atributo o una estructura cualquiera. Debido a esto, hay zonas del fichero de presentación visual en las que se repite un mismo *byte* un número determinado de veces.

La segunda característica de la que vamos a hacer uso es que, por complicado que sea el dibujo, prácticamente siempre hay algún *byte* que no es utilizado. Hemos comprobado esto en bastantes pantallas (e incluso programa en máquina o la propia ROM) y siempre ha sido así.

Sistema de codificación

La rutina COMPRF (Listado 1) es la encargada de COMPRimir las Pantallas. Se localiza junto con DCMPRP, en el buffer de la ZX-Printer, por lo que no debe utilizarse el canal «P» mientras las usemos. Debe ser introducida con la dirección de memoria donde ha de almacenarse la pantalla comprimida en el par HL; en el listado es 50.000 (línea 130), pero puede modificarse siempre que quede por encima de la RAMTOP (un valor aconsejable para los usuarios de Spectrum 16K puede ser 26.000).

Tras almacenar HL en el *stack* y poner a cero del registro A, se ejecuta la subrutina LOCPRF, encargada de LOCalizar el número que será utilizado como PReFijo para codificar las zonas de pantalla en las que un *byte* se repita 3 o más veces. Para esto utilizamos la instrucción CPIR, que a la salida dará Z activado sólo si el valor que carga A ha sido localizado en la pantalla. Si no se ha encontrado, este es el número que necesitamos, por lo que se salta a COMPR1. Pero si se encuentra, damos un nuevo valor a A y se repite el proceso, a no se que A vuelva a valer cero, lo que querría decir que ya se ha probado con todos los valores posibles, por lo que se saldría con un error «Out of DATA» para indicar que no es posible codificar esa pantalla (en la práctica sólo sucedió esto cuando se había preparado específicamente la pantalla para que ocurriera así).

Una vez pasada esta fase, se almacena como primer *byte* del fichero comprimido el número que va a utilizarse como prefijo. Después, comienza a copiarse el fichero de pantalla en la nueva localidad, comprobando si un valor se repite dos veces y llamando en ese caso a CPREP. Además, se comprueba si HL vale cero, lo que indicaría que hemos dado una dirección de destino demasiado alta, y asi es así, da error «Out of memory» (Para el 16K deberá sustituirse el XOR A de la línea 420 por LD A,128). También se comprueba si el *byte* más significativo de IX es 5Bhex, lo cual significaría que se ha codificado la totalidad de la pantalla y por lo tanto el fin de la rutina, por lo que se retorna dando a BC el valor de HL (dirección del *byte* que sigue al fin de la pantalla comprimida). Para conseguir cargar en el acumulador el registro I (de IX)



no nos ha quedado más remedio que recurrir a un pequeño truco: utilizar la instrucción equivalente en el par HL antecedida de DDhex, lo que, debido a la arquitectura usada en el X-80 para la ampliación del juego de instrucciones, equivale a LD A,I (I de IX y no el registro de interrupciones).

Por último, CPREP ComPrar los sucesivos *bytes* con A y REPite incrementando B hasta que alguno sea distinto o B llegue a cero (=256), en cuyo caso almacena el prefijo seguido del número de veces que se repite el valor de A, y retorna para que A sea almacenado y continúe COMPR2.

Expandiendo las pantallas comprimidas

La rutina DCPRMP (Listado 2), localizada tras la anterior, permite DesCoMPRimir las Pantallas que hubieran sido codificadas. A la derecha, HL debe contener la direc-



ción de inicio del fichero que queramos descomprimir (50.000 en el listado). El funcionamiento es sencillo: se carga en el registro C el prefijo y se comprueba si los siguientes números del fichero son iguales a éste. Si no son iguales, se pasan a pantalla normalmente, pero, si alguno lo es, se salta a DCMPR3, donde se cargan los dos valores siguientes y, mediante un bucle, se realiza la expansión. En el bucle principal se comprueba el registro H, para retornar cuando el valor 5Bhex lo indique.

Nuevo formato

Así pues, el principio de un fichero «comprimido» podría ser:

251, 13, 50, 251, 100, 0, 251, 0, 1,...

Lo que quiere decir que el número usado como prefijo es el 251, por lo que al descomprimirlo quedaría primero un 13 y un 50; después vemos que aparece el prefijo, por lo quel siguiente número es la canti-

LISTADO 3

```
   1 REM Carg. COMPRESOR PANT.
  10
  20 PRINT  FLASH 1;"Pokeando"''
  30 FOR N=0 TO 15
  40 LET C=0
  50 READ A$
  60 FOR M=1 TO 14 STEP 2
  70 LET L=FN D(A$(M TO M+1))
  80 POKE 23296+N*7+INT (M/2),L
  90 LET C=C+L
 100 NEXT M
 110 READ B
 120 IF C<>B THEN  PRINT  FLASH
1;" ERROR EN LINEA ";200+N*10;"
"
 130 NEXT N
 140
 200 DATA "2150C3E5AF2100",745
 210 DATA "4001001BEDB120",538
 220 DATA "053D20F3CF0DDD",782
 230 DATA "210040E177234F",555
 240 DATA "DD7E00DDBE01CC",963
 250 DATA "385B7723AFBC20",696
 260 DATA "02CF03DD23DD7C",813
 270 DATA "FE5B20E7E5C1C9",1231
 280 DATA "DDBE02C00602DD",834
 290 DATA "23DD23042805DD",561
 300 DATA "BE0128F6712370",737
 310 DATA "23C92150C31100",561
 320 DATA "404E2379FE5BC8",844
 330 DATA "7EB928050CEDA0",765
 340 DATA "18F32346237E12",551
 350 DATA "1310FC18E88000",671
 360
 500 DEF FN P(H$)=CODE H$-48-7*(
CODE H$>64)
 510 DEF FN D(H$)=FN P(H$(1))*16
+FN P(H$(2))
```

dad de veces que se repite el que viene a continuación, es decir que tras el 50 vendrían 100 ceros. Seguiría un 10 y de nuevo 256 (=0) unos.

En este ejemplo el ahorro de memoria sería grande: 10 *bytes* frente a 359, pero en pantallas de carga normales suele variar entre un 25% en las más complejas y un 60 o 70% en las sencillitas. Esto no es nada despreciable si tenemos en cuenta que un 25% de una pantalla son casi 1.7K de memoria, lo cual supera ampliamente los 110 *bytes* que ocupan las dos rutinas juntas.

Cargador Basic

Quienes no dispongan de ensamblador ni de conocimientos de lenguaje máquina no deben desesperar, también pueden utilizar estas rutinas. Para ello deberán cargar el programa BASIC del Listado 3 al pie de la letra y ejecutarlo.

En caso de que sea detectado algún error en las líneas DATA, en cuyo caso se dará un mensaje indicando la línea en que esta, habrá que corregirlo y ejecutar de nuevo el programa. Cuando no haya errores, podrá ser salvado a cinta en forma de *bytes* con SAVE «nombre» CODE 23296,110, y cargarlo posteriormente con LOAD " " CODE.

Para comprimir la pantallal en memoria debe hacerse LET L=USR 23296, lo que la codificaría dejando en la variable L la dirección del *byte* posterior al final del fichero. Esto la almacenará a partir de la dirección 50.000, por lo que podemos averiguar la longitud de la pantalla comprimida haciendo PRINT L−50000. Si queremos situar el fichero comprimido en un lugar determinado de la memoria (muy aconsejable que sea encima de RAMTOP) deberemos pokear la dirección escogida como sigue:

```
LET H=INT (dirección/256)
POKE 23297,dirección−H*256:
POKE 23298,H
```

Por ejemplo, para un Spectrum de 16K, si queremos que comience a cargar a partir de la dirección 26000, habrá que hacer:

```
CLEAR 25999: LET H=
INT(26000/256):
POKE 23297,26000−H*256: PO-
KE 23298,H
```

Si acabamos de ejecutar la rutina para otra pantalla, podemos sustituir dirección por la variable L, con lo cual se almacena inmediatamente después de la anterior.

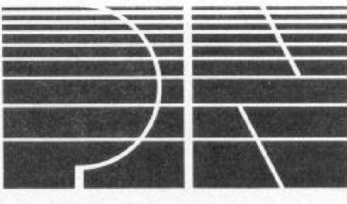
Para expandir un fichero cargado anteriormente habrá que pokear la dirección de inicio (sólo si es distinta de 50000) en 23376 y 23377 como anteriormente, y hacer RANDOMIZE USR 23375.

LISTADO 2

```
760
770 ;-----------------------
780 ; DesCoMPRimir Pantalla
790 ;
800 ;ENTR: HL=INICIO FICHERO
810 ;-----------------------
820
830
840         ORG   23375
850
860         LD    HL,50000
870 DCMPRP
880         LD    DE.16384
890         LD    C.(HL)
900 DCMPR1
910         INC   HL
920 DCMPR2
930         LD    A.D
940         CP    ^5B
950         RET   Z
960         LD    A,(HL)
970         CP    C
980         JR    Z,DCMPR3
990         INC   C
1000        LDI
1010        JR    DCMPR2
1020 DCMPR3
1030        INC   HL
1040        LD    B,(HL)
1050        INC   HL
1060        LD    A,(HL)
1070 DCMPR4
1080        LD    (DE),A
1090        INC   DE
1100        DJNZ  DCMPR4
1110        JR    DCMPR1
```

LISTADO 1

```
10 ;--------------------
20 ; (c) Patatita soft
30 ;--------------------
40 ; COMPRimir Pantalla
50 ;
60 ;ENTR: HL=DESTINO
70 ;SALD: BC=FIN FICHERO
80 ;--------------------
90
100
110         ORG   23296
120
130         LD    HL,50000
140 COMPRP
150         PUSH  HL
160         XOR   A
170 LOCCOD
180         LD    HL,16384
190         LD    BC,6912
200         CPIR
210         JR    NZ,COMPR1
220         DEC   A
230         JR    NZ,LOCCOD
240
250 ;         "E Out of DATA"
260
270         RST   8
280         DEFB  13
290 COMPR1
300         LD    IX,16384
310         POP   HL
320         LD    (HL),A
330         INC   HL
340         LD    C,A
350 COMPR2
360         LD    A,(IX)
370         CP    (IX+1)
380         CALL  Z,CPREP
390         LD    (HL),A
400         INC   HL
410         XOR   A
420         CP    H
430         JR    NZ,COMPR3
440
450 ;         "4 Out of memory"
460
470         RST   8
480         DEFB  3
490 COMPR3
500         INC   IX
510         DEFB  ^DD
520         LD    A,H
530         CP    ^5B
540         JR    NZ,COMPR2
550         PUSH  HL
560         POP   BC
570         RET
580 CPREP
590         CP    (IX+2)
600         RET   NZ
610         LD    B,2
620         INC   IX
630 CPREP1
640         INC   IX
650         INC   B
660         JR    Z,CPREP2
670         CP    (IX+1)
680         JR    Z,CPREP1
690 CPREP2
700         LD    (HL),C
710         INC   HL
720         LD    (HL),B
730         INC   HL
740         RET
```


Paseo de Gracia, 11
Esc. C., 2.º 4.º

Tel. (93) 318 24 53
08007 Barcelona

PIN SOFT, S.A.

NOVEDADES SPECTRUM



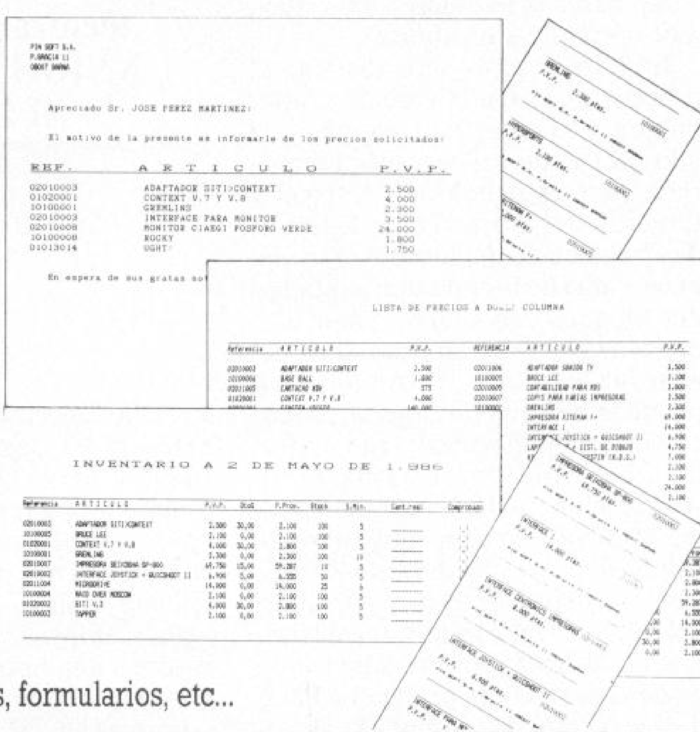
I/F JOYSTICK 2.500

— Con paro de imagen, para que puedas seguir el juego, paso a paso.



Consulta nuestro extenso catálogo

JOYSTICK + Interface incorporado 3.750



ALIMENTACION ININTERRUMPIDA 9.750

— Para que los cortes de suministro eléctrico no te dejen a oscuras, l'30 h. de autonomía.

SITI-PRINTER 4.480

— Aprovecha tu impresora al cien por cien.
— Listados de hasta 132 columnas, recibos, impresos, formularios, etc...
— Utiliza la base de datos SITI.

TIENDA AL PUBLICO EN BARCELONA

PEDIDOS POR CORREO O TELEFONO

Envíos contra reembolso a toda España
200 ptas. gastos de envío
En tu domicilio en 3-4 dias

Nombre ______________________
Dirección ______________________
Población ______________________
Pedido ______________________

Enviar a: PIN SOFT, S.A. - P.º de Gracia, 11 - Esc. C, 2º 4º - 08007 Barcelona

SWEEVO'S WORLD

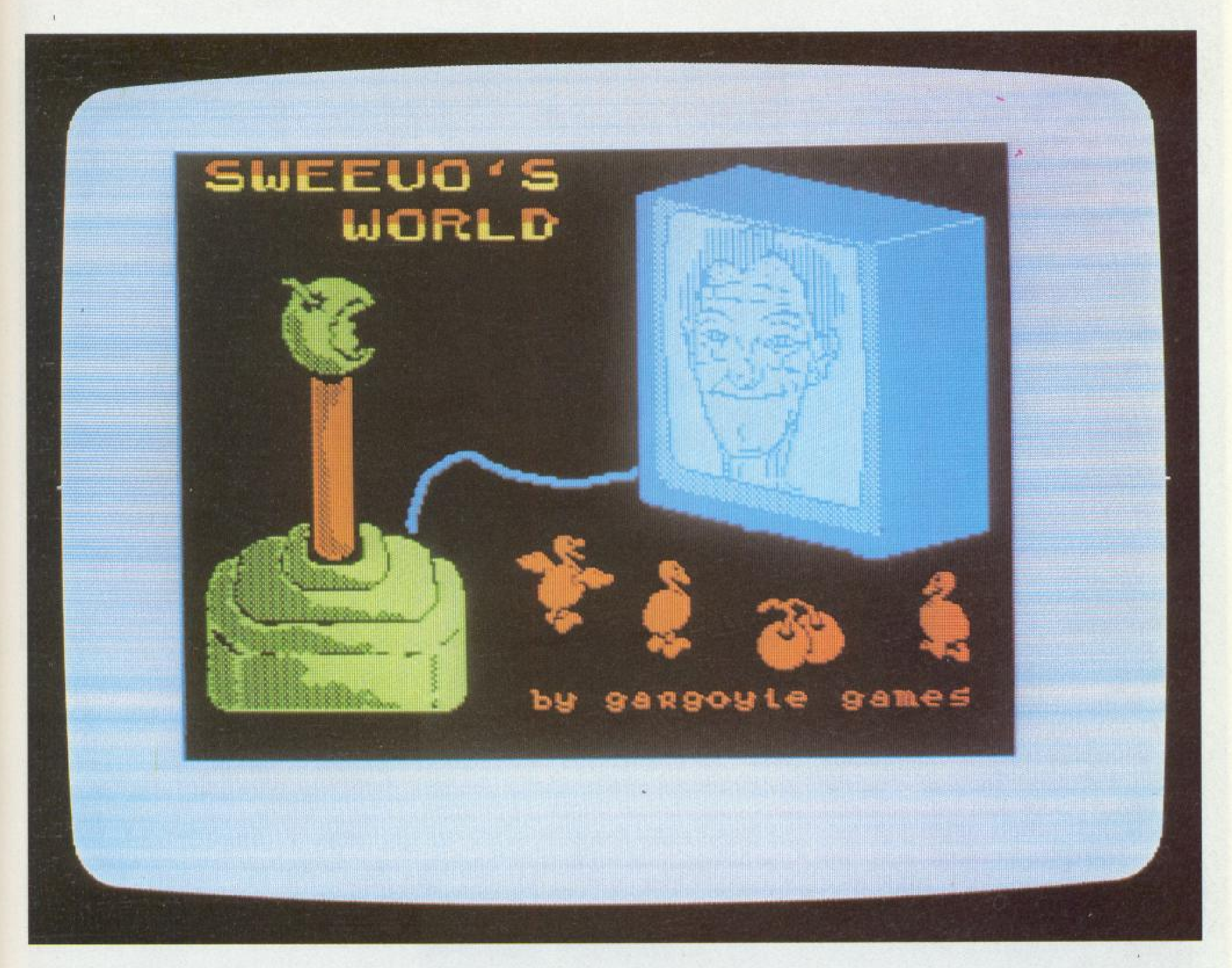
Este mes nos enfrentamos en nuestra guía de hackers con el programa SWEEVO'S WORLD. Está realizado en tres dimensiones al estilo de los dos primeros programas en que ULTIMATE utilizó la técnica filmation: KNIGHT LORE y ALIEN 8.

En esta ocasión voy a contar la historia de los POKEs al mismo tiempo que los busco, con lo cual de momento lo único que puedo decir es que aparentemente el programa se carga sin una protección excesivamente complicada. Todo parecen ser bloques a velocidad normal y sin cabecera.

El primer paso será rastrear el proceso de carga. Cargo el primer bloque y lo paro. Nada se opone a ello. El listado del cargador no aparece, pero es debido a que papel y tinta son del mismo color. Después de arreglar esto, podemos ver lo que hace: carga de la cinta una pantalla y dos bloques con CODE, y por último lanza el programa en la dirección 24800. Anulo esto último y dejo que se ejecute. La carga se produce sin problemas acabando con el informe 0 OK. Pruebo a lanzarlo en su dirección y el juego comienza correctamente. No tiene ningún truco. Además parece que durante la carga deja sitio suficiente para el buffer del microdrive, con lo que, teóricamente, la operación de adaptación no debe dar ningún problema. Pasamos a averiguar la longitud y direcciones de los dos bloques en que esta almacenando el juego. Esto se puede hacer de muchas maneras. Yo utilizo uno de los múltiples programas para leer cabeceras. El resultado es que el primer bloque empieza en la dirección 64400 y tiene 1085 bytes, y el segundo empieza en 24800 y tiene 33456 bytes. No hay ningún problema, para pasarlo a microdrive. Incluso parece que queda en medio, en la dirección 58256, suficiente espacio para el monitor.

Para empezar a analizar vamos a la dirección en que se lanza el programa. Nos encontramos con un salto a # 9CD2. Aquí vemos que modifica algunas variables del sistema BASIC refiriéndolas al registro IY. Esto parece indicar que se va a apoyar en la ROM para algunas funciones. En principio su supone que lo hará para la impresión de mensajes, puesto que ha abierto el canal 1. Por lo demás, aquí la zona de memoria comprendida entre # A708 y # AB07, utilizando para ello el registro IX. Al final llama a una subrutina en la dirección # 908B, pero más que eso debe ser el programa principal, ya que si volviera la siguiente instrucción es un salto a la dirección # 0000 que destruirá el juego.

Pasamos a la dirección # 908B. Casi al principio nos encontramos con un sugestivo LD A, # 05. Debe ser el número de vidas, pero antes de comprobarlo vamos a echar un vistazo a lo que hay detrás. Se inicializan un montón de variables, se llama a algunas subrutinas y se cierra un bucle sobre la dirección # 90E3 despueśs de comprobar el dato contenido en # 6176 y si están pulsadas simultáneamente las teclas SYMBOL-SIFT y 5. Esto sugiere muchas cosas y, aunque es posible que alguna no sea cierta, vamos a contarlas. Todo lo que hay entre las direcciones # 908B y # 90E3 debe corresponder con el menú que aparece al principio de la partida, en el que se nos da a elegir entre los cuatro niveles planeta. La dirección # 6176 debe controlar si nos han matado definitivamente y hay que empezar una nueva partida. Las instrucciones deben estar equivocadas cuando dicen que para el juego hay que pulsar SS + 0, ya que todo indica que se hace



con SS + 5. Por último, al principio de lo que se supone el bucle de juego, hay un bucle en el que se espera la llegada de tres interrupciones. Es de suponer que modificando este valor se podrá variar la velocidad del juego.

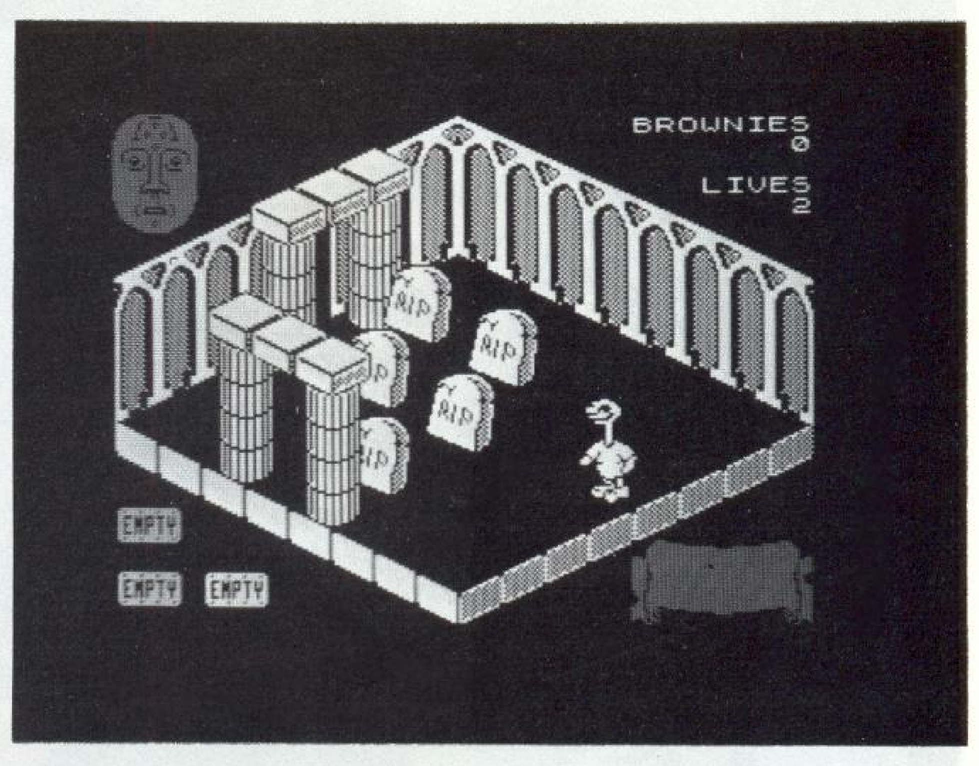
Para tratar de comprobar todo esto realizamos algunas modificaciones y lanzamos el programa. Incremento lo que parece el número de vidas y aumento la cantidad de interrupciones que debe esperar. Parece que hemos acertado, el juego es más lento, tenemos más vidas y pulsando SS + 5 se para.

Volvemos a cargar el programa y continuamos investigando. Sabiendo que en la dirección # 611A se guardan las vidas que nos quedan, no debemos tener problemas para encontrar el POKE de vidas infinitas. Buscamos cuántas veces es utilizada esta variable y la encontramos cinco veces. Una para inicializarla al principio de la partida, otra para escribirla en la pantalla, otra para comprobar si nos quedan más vidas y dos más para decrementarla (una para leer el antiguo valor y otra para poner el nuevo). Con esto ya tenemosf las vidas infinitas, basta con poner en la dirección #81C3 el dato # A7 (sustituimos la instrucción DEC A por AND A). Otra cosa que también se supone aunque todavía no la he comprobado es que en la dirección # 6119 se guarda el estado del robot. Durante la inicialización se le da el valor 4 y se maneja conjuntamente con el número de vidas. Para evitar que decrezca probamos a poner en #835A un #B6 (cambiar DEC (HL) por OR (HL)).

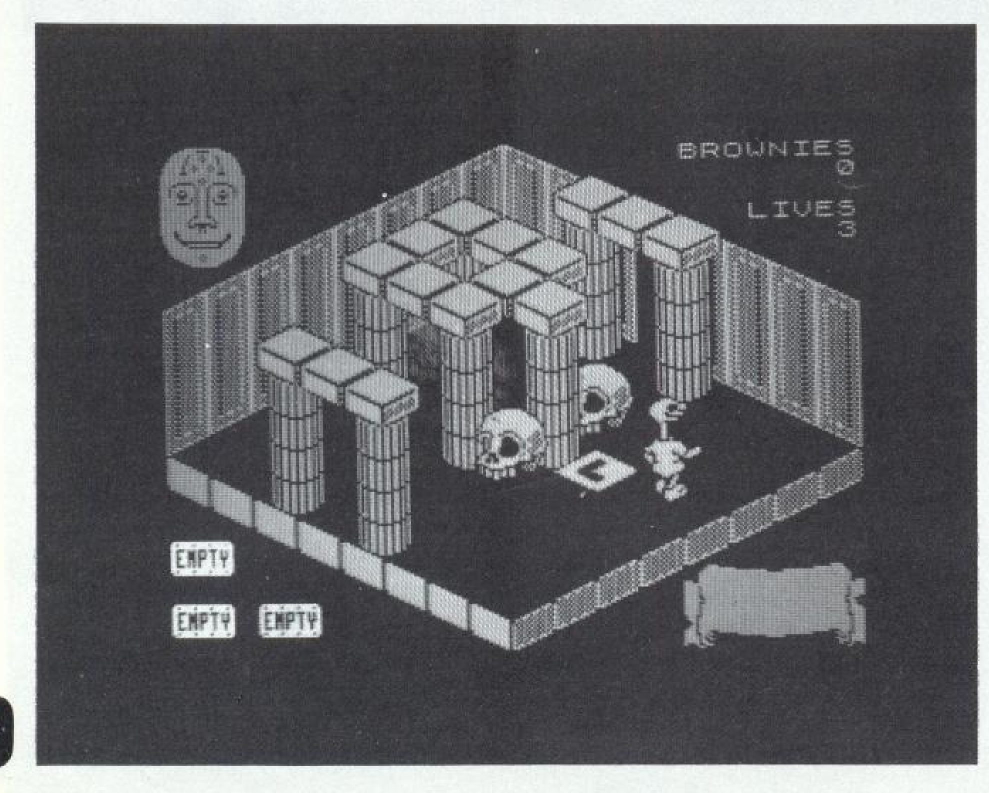
Acabo de probar los dos POKEs anteriores y funcionan bien. Con ellos ya se puede abordar el juego con bastantes probabilidades de llegar al final en la primera partida, pero todavía es demasiado difícil y nuestro análisis no debe de quedar aquí. El próximo paso será tratar de averiguar qué rutinas son las encargadas de controlar los gráficos y del movimiento de los objetos. También intentaremos descubrir cómo está codificado todo el juego.

Antes de entrar en el bucle de juego hay tres llamadas a subrutinas que en teoría se deben escargar de inicializar variables y del menú de presentación. Pruebo a parar el programa después de la primera llamada para comprobar qué cambios ha habido pero no es posible volver al monitor después de la rutina. La zo-

na de memoria donde está el monitor debe ser utilizada para algo. Busco cuándo se refiere a ella y me encuentro con que se usa para la creación de los gráficos. Tiene exactamente 6K, lo justo para almacenar toda la pantalla sin atributos. Todos los gráficos se montan en esta zona y luego llamando a la rutina # 747B se **vuelcan** sobre la pantalla. Con esta forma de actuar consiguen varias ventajas. La organización de la memoria a la hora de pintar un gráfico es la lógica en la que a continuación de la representación de una línea de pantalla se encuentra la correspondiente a la línea inmediatamente inferior. De esta forma su utilización es bastante más sencilla y solamente hay que tener cuidado a la hora de volcar sobre pantalla. Por otra parte, se pueden despreocupar del problema de que al borrar un gráfico para ponerlo en otra posición se pueda ver un momento en que no está en pantalla ninguno de los dos. A cambio, el mover 6K de memoria cada vez que se modifica algo requiere un tiempo, pero está bien resuelto para que sea lo más rápido posible y no frene el juego.



Toda esta instrucción, aunque no nos permita directamente encontrar ningún POKE, seguramente será muy útil para poder comprender el resto del programa. Para empezar descubrimos que la primera de las tres llamadas (# 7B0D) se encarga de pintar todos los gráficos fijos en una pantalla. Justo delante hay un tentador LD A, # FE que no puedo resistirme a cambiar. Lanzo el programa y el menú de presentación aparece sobre una de las pantallas del juego. Con esto ya podemos intentar escribir una rutina que nos muestre todas las habitaciones del planeta. El único problema es en qué dirección colocarla. El *buffer* de la impresora no nos sirve por que es usado por el programa, tampoco podemos destruir las variables del sistema BASIC, así que de momento utilizaremos la zona correspondienteds al programa en BASIC, aunque no tenemos demasiado sitio.



La primera prueba para ver todas las pantallas ha resultado un fracaso. Nada más lanzar el programa se queda colgado. Habrá que seguir investigando lo que ocurre en la dirección # 7B0D. Las pantallas están codificadas a partir de la dirección # 6278 y para ahorrar memoria cada una sólo ocupa los *bytes* necesarios para su definición. Las pantallas que no existen (los huecos del mapa) no gastan ni un *byte.* El primer dato de cada pantalla indica el número de *bytes* que utiliza su definición, el segundo es el número de pantallas. El mapa será organizado en cuatro bloques de 8*8 pantallas, en total 256 posibles habitaciones, pero no todas existen. Algunas son utilizadas para

TABLA I

POKE 33219,167	Vidas infinitas
POKE 37880,X	Número de vidas
POKE 34236,X	Tiempo para cambiar objeto
POKE 37099,X	Velocidad
POKE 33626,182	No cambia de estado
POKE 35353,142	No nos matan los enemigos
POKE 36215,201	Tablas quietas
POKE 30263,33	Quita las tablas
POKE 36324,0	Quita los dedos
POKE 32161,24	Evita muerte parcial por el dedo
POKE 32219,142	Evita muerte total por el dedo
POKE 32034,134	Evita muerte parcial por los objetos fijos
POKE 32007,142	Evita muerte total por los objetosd fijos
POKE 32874,33	Sigue teniendo los objetos cuando los deja
POKE 35728,110	Evita la destrucción de los objetos por el dedo
POKE 35786,0	Se puede matar los enemigos con cualquier objeto
POKE 30238,24	Transforma los muñecos del suelo en dedos
POKE 33437,104	Evita la muerte por la pesa pero te atrapa
POKE 36089,176	Evita quedar atrapado por la pesa
POKE 33442,104	Inmortalidad total

PROGRAMA 1

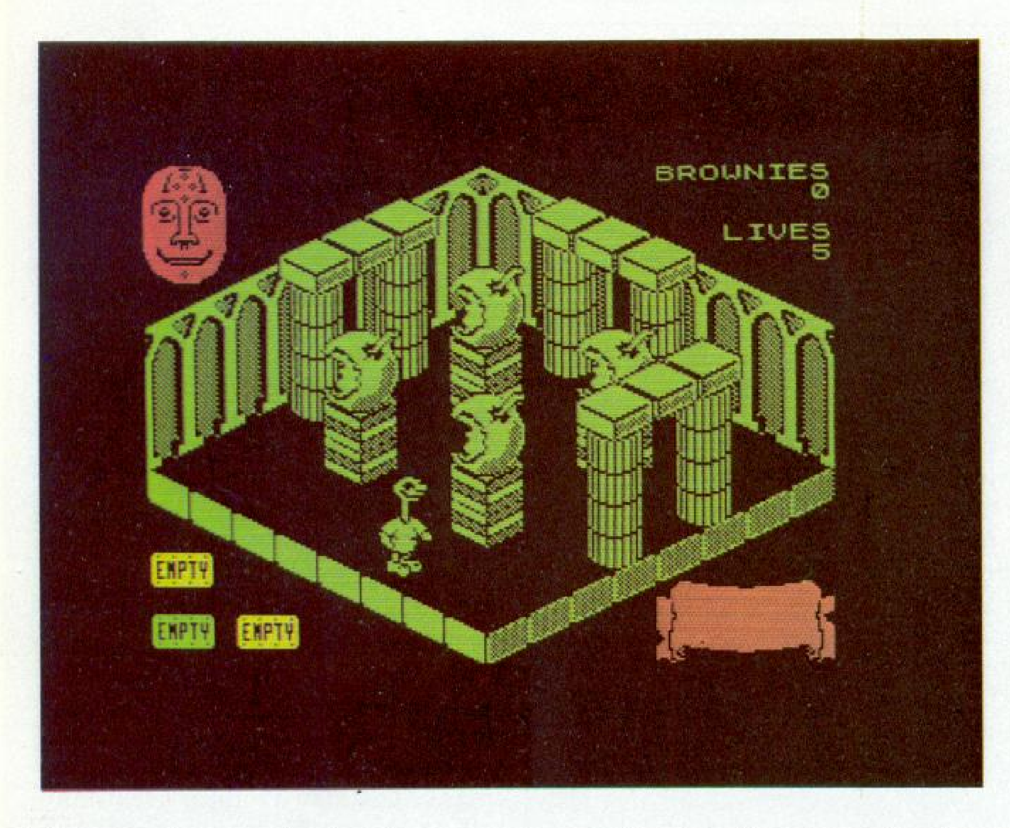
```
  10 CLEAR 60000: LET n=58256:
RESTORE
  30 FOR i=1000 TO 1020 STEP 10
  40 READ a$,a: LET s=0
  50 FOR j=1 TO LEN a$-1 STEP 2
  60 LET d=16*(CODE a$(j)-48-7*(
a$(j)>"9"))+CODE a$(j+1)-48-7*(a
$(j+1)>"9")
  70 POKE n,d: LET n=n+1: LET s=
s+d: NEXT j
  80 IF s<>a THEN PRINT "Error
en la linea ";i: STOP
  90 NEXT i
 100 LOAD ""SCREEN$ : RANDOMIZE
USR 58256
1000 DATA "DD2190FB113D043EFF37C
D560530F1DD21E06011B0823EFF37CD5
60530F121C2",3769
1010 DATA "E311055D012C00EDB0210
55D22D590C3E06021786222315D2A315
D7E5F160023",2726
1020 DATA "7E2BFEFE28EC1922315DC
D0D7BCD998FCD7B742108SC36007EA72
8FC18DA",3454
```

los menús de principio y final del juego. El problema que tenía la rutina para imprimir todas las pantallas era que intentaba pintar las que no existen. Con esto ya estamos preparados para escribir una nueva versión que funciona perfectamente. La tenéis en el programa 1.

Ya que estamos en ello vamos a seguir analizando cómo está codificado el mapa. Para extraer la información de la tabla llama a la dirección # 76A9, donde vemosf que el tercer *byte* tiene la información de las cuatro posibles salidas laterales de la habitación. Por eliminación el cuarto *byte* debe contener el color del fondo y los gráficos del fondo. El *bit* 7 de este *byte* indica si la habitación ha sido recorrida o no para poder calcular el tanto por ciento de aventura. Estos 4 *bytes* son los únicos indispensables (corresponderían a una habitación vacía). Los demás se destinan a indicar la existencia de objetos. Cada uno de estos se codifica en dos *bytes* en los que el primero indica la posición dentro del mapa 8*8 en que está dividida cada habitación, y el segundo el tipo de objeto de que se trata. De esta forma la cantidad de memoria necesaria para representar las pantallas es bastante pequeña y se puede incluir gran número de ellas, todas distintas.

En este punto sí quisiéramos seguir con el análisis metódico deberíamos tratar de averiguar cómo están codificados los objetos no fijos (bota, lata, oso), las trampas móviles (dedo, etc.) y los enemigos, sin embargo, vamos a intentar una nueva estrategia. Partiendo del punto en que se decrementan las vidas intentaremos ir hacia atrás en el programa hasta encontrar dónde se realiza el test de colisión con los objetos mortales. De esta forma se suele llegar más rápidamente a los POKEs, pero se corre el riesgo de caer en un callejón sin salida y perder mucho tiempo. Además, no es muy aconsejable utilizar esta técnica desde el principio, ya que en muchos casos se actúa por intuición y para no equivocarse conviene tener una

GUIA DEL HACKER



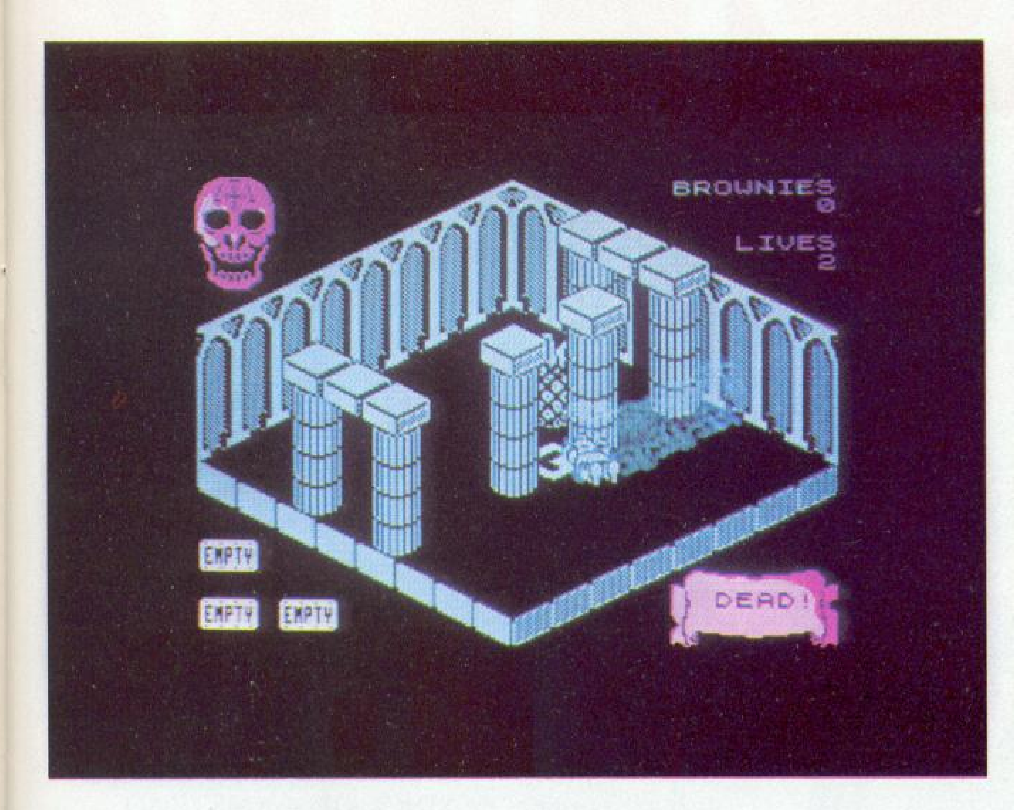
idea de cuáles son las variables del sistema.

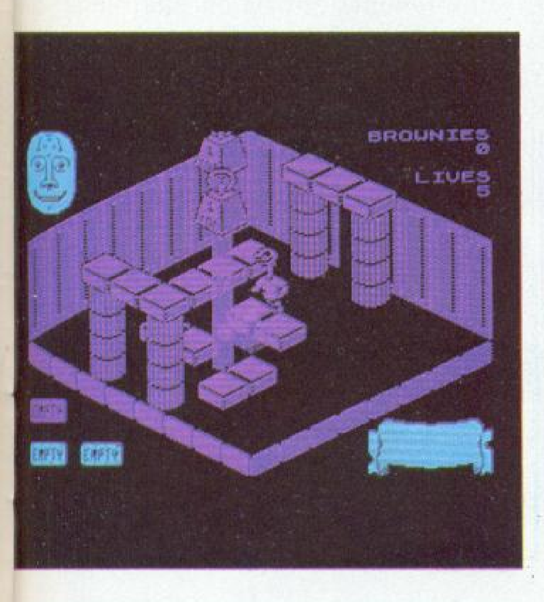
El número del sistema se decrementa # 81C3. Vamos un poco hacia atrás y nos encontramos con que la rutina empieza en # 81AF. Buscamos las llamadas a esta dirección que hay en el programas y encontramos dos. Retrocediendo desde la primera y después de buscar qué salto relativo va a parar la dirección # 834E encontramos lo que buscamos, o por lo menos eso parece. Hay una llamada a la dirección # 91A3 y después se testean varios *bits* de un dato referido como (IX+13). En función de estos *bits* se realizan saltos que traen como resultado la pérdida de una vida.

Sin embargo, al mirar la rutina a la que llama nos encontramos con que no tiene nada que ver y se limita a la lectura del teclado o *joystick* según corresponda. De todas formas,

PROGRAMA 2

```
  10 CLEAR 60000: LET n=58256: R
ESTORE
  30 FOR i=1000 TO 1110 STEP 10
  40 READ a$,a: LET s=0
  50 FOR j=1 TO LEN a$-1 STEP 2
  60 LET d=16*(CODE a$(j)-48-7*(
a$(j)>"9"))+CODE a$(j+1)-48-7*(a
$(j+1)>"9")
  70 POKE n,d: LET n=n+1: LET s=
s+d: NEXT j
  80 IF s<>a THEN  PRINT "Error
en la linea ";i: STOP
  90 NEXT i
 100 LOAD ""SCREEN$ : RANDOMIZE
USR 58256
1000 DATA "DD2190FB113D043EFF37C
D560530F1DD21E06011B0823EFF37CD5
60530F121D6",3789
1010 DATA "E311055D01F800EDB011C
0FF013100EDB021055D22D59021C0FF2
2FF9021E060",3719
1020 DATA "22D79CC3E060CD019C219
0E31191E301FF09EDB0CD85981603071
00756696461",3686
1030 DATA "7320696E66696E6974617
35ECDC25D21C381363D380236A7CD859
8160507496E",3156
1040 DATA "6D2E20746F74616C5ECDC
25D21A282364F3802366BCD859816070
7496E6D2E20",2891
1050 DATA "706172636961 6C5ECDC25
D21A17D11227D01C6283803018619 70E
B71CD859816",3242
1060 DATA "0907496E6D2E207065736
15ECDC25D219D8211F98C01006F38030
1B06870EB71",3035
1070 DATA "CD8598160B07536 96E207
461626C61735ECDC25D2137763 6CD380
236213E0432",2803
1080 DATA "1F62C9AF32085C3A1A62E
60720083A1F62EE02321F62CD7B74CD6
B97CD998F3A",3186
1090 DATA "085CF620FE73280DFE6E2
0DBCD85982 04E4F5E37C9CD85982053 4
95EA7C93EF7",3893
1100 DATA "DBFE1F3813CD7694AF32E
B60DD2114622 13A62CBEEC3AB8F1FD28
E811F380DDD",3950
1110 DATA "21004011001B3EFFCDC60
4FBC3998F",1607
```





PROGRAMA 3

```
  10 BORDER 0: PAPER 0: INK 0
  20 CLEAR 24799
  30 LOAD ""SCREEN$
  40 PRINT AT 3,0;: LOAD ""
     CODE
  50 PRINT AT 3,0;: LOAD ""
     CODE
 100 REM comienzo de lo pokes
1000 RANDOMIZE USR 24800
```

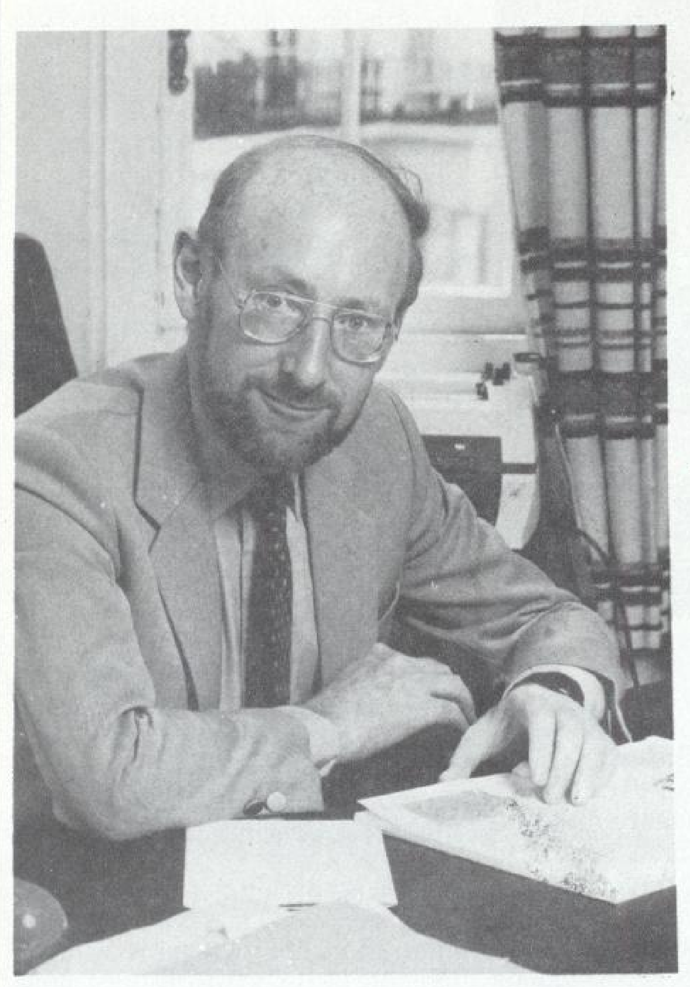
vamos a comprobar para qué sirve cada uno de los saltos. Para ello, anularemoss uno de ellos cada vez y vemos frente a que somos inmortales. El primero de ellos se realiza después de testear el *bit* 5 y anulándolo conseguimos que las pesas no nos aplasten en su caída. Pero cuando nos caen encima, Sweevo queda atrapado bajo su peso con lo que si no encontramos otra solución será mejor que nos mate. El segundo salto corresponde al *bit* 2 y nos concede la inmortalidad contra todo lo que antes nos hacía perder una vida (excepto la pesa). El tercer salto es con el *bit* 0 y elimina la muerte parcial. Sin embargo, tiene el problema de que al acabar la partida nuestro héroe se pone a hacer tonterías y no se puede empezar de nuevo. Habrá que arreglar eso. El último salto parece que controla cuando no caemos por un agujero en vista de los efectos que produce. Si retrocedemos un poco nos encontramos con que la rutina empieza en # 827F y no existe en este trozo nada que pueda modificar alguno de estos *bits*. Por lo tanto, debe de estar en otro rutina o en la misma más adelante. Siguiendo por este camino nos encontramos con dos llamadas a las direcciones # 7E28 y # 7ED5. Ahí seguirá nuestra búsqueda. Entre otras cosas llama repetidas veces a la dirección # 7C75, y es aquí donde manipula repetidas veces los datos que nos interesan. Después de analizarla se llega a la conclusión de que se encarga de controlar las colisiones con los objetos fijos (en los que se incluye el resorte del ascensor y el muñeco que surge del suelo una vez que esta fuera. También se llama a la dirección # 7D2B que hace las mismas comprobaciones respecto a las trampas móviles.

Todos los POKEs útiles encontrados los tenéis en la tabla 1. Para usarlos teclear el programa 3 y poner a partir de la línea 100 los que queráis. Por último lanzad el programa y cargar la copia original. El programa 2 es un cargador con los POKEs más interesantes, que incluye tres funciones nuevas. Apretando la tecla 1 se consigue matar a cualquier enemigo. La 2 nos permite abortar el programa y la 3 saca una copia de pantalla.

Para acabar diremos que se trata de una pequeña maravilla de la programación. El programa está perfectamente estructurado, lo que facilita mucho nuestra tarea de búsqueda. Hay que destacar la curiosa forma en que maneja los bloques de datos correspondientes a los objetos móviles. Los tiene organizados en forma de listas al estilo del PASCAL. Cada bloque tiene almacenado un puntero que señala al siguiente. Existen dos listas de este tipo, una con la información de los objetos «vivos» y otra con los bloques que no se están utilizando y que están listos para la creación de cualquier objeto. También hay que decir que la memoria está bastante aprovechada. Hubo problemas para localizar un sitio donde entrara el cargador con los POKEs. Finalmente, se cambió la posición de la pila.

MANUEL ARANA

NOTI

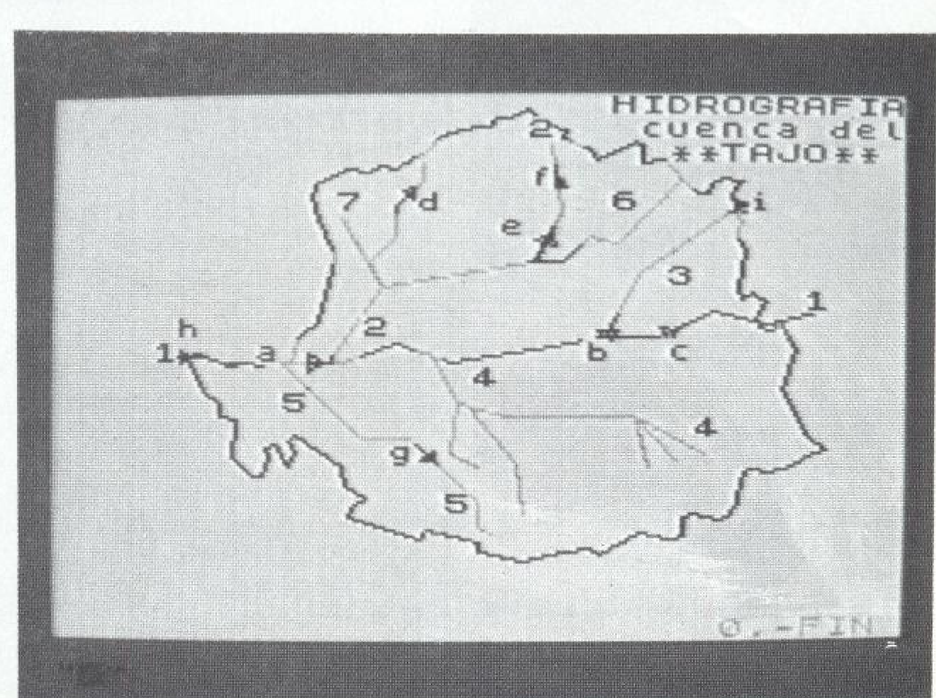


¡Amstrad Consumer Electronics adquiere Sinclair Research!

Sir Clive Sinclair y Alan Sugar, presidentes respectivamente de SINCLAIR y AMSTRAD, comunicaban la sorprendente noticia en una rueda de prensa celebrada el lunes 7 de abril.
Sugar anunció la adquisición de los derechos de los ordenadores Sinclair por un importe de cinco millones de libras.

Software extremeño para la educación

El ICE de la Universidad de Extremadura ha hecho públicos los ganadores del I Concurso de Programas Didácticos para Ordenadores, dentro de los actos realizados con motivo de la VIII Semana de Extremadura en la Escuela. El primer premio ha sido otorgado a un programa realizado por Santiago Hdez. Cano y Angel Prieto para alumnos de segunda etapa de EGB compuesto de tres bloques («Observa y comenta», «Pregunta lo que ignores» y «Demuestras lo que sabes») que facilitan el aprendizaje de la geografía extremeña.

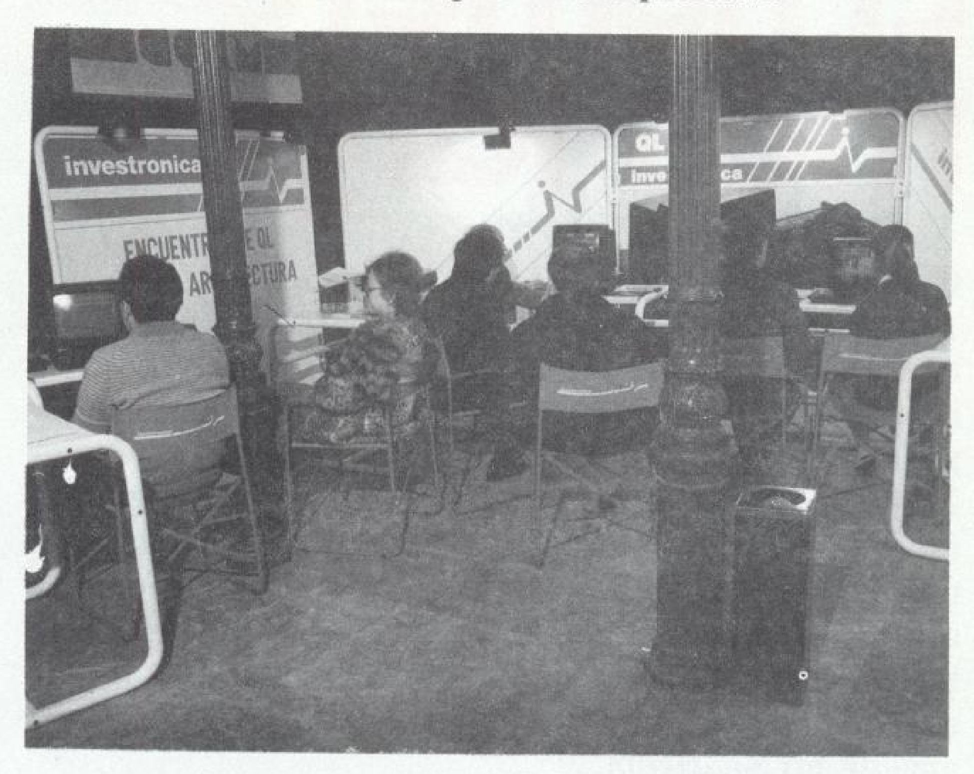


CIAS

Novedades de Talent

Talent Computer System ha lanzado recientemente dos nuevos productos para el QL, The Assembler Workbench y «TechniQL». El primero es un conjunto de utilidades para la escritura edición y depuración de programas en lenguaje ensamblador, formado por tres módulos principales: editor de pantalla completa, ensamblador y monitor desensamblador. El segundo es un paquete de CAD complementario del famoso GraphiQL. TechniQL permite la creación de complejos dibujos de tamaño varias veces mayor que la pantalla.

Encuentros de QL con la arquitectura



Organizadas por Investrónica se han celebrado en EXPO-SITAP, Colegio Oficial de Arquitectura, los Primeros Encuentros de QL con la Arquitectura.
Durante estas jornadas se realizaron demostraciones prácticas sobre la utilidad y nuevas posibilidades que ofrece la informática aplicada a la Arquitectura.

Domenech, Informática Educativa

Preocupada por la inclusión de la informática en el mundo educativo, la empresa catalana Domenech ha lanzado un nuevo título editorial.
Con el nombre de «La tortuga va a la EGB», presenta un cuaderno de fichas y actividades para el ciclo medio y superior de EGB, que no dudamos alcanzará el objetivo de difundir el lenguaje Logo entre los escolares.

RA-MA, nuevos títulos

68000, GUIA DEL USUARIO de Lionel Fleetwood es uno de esos libros imprescindibles para cualquiera que desee comprender los misterios del lenguaje ensamblador, y pretenda aplicarlo a proyectos reales.
Con un precio de 1.980 ptas. es una de las novedades que presenta en este trimestre la editorial RA-MA.

Factores primos:

El número más largo



Cuando leí las bases del concurso matemático de Todospectrum decidí participar, pensando que sería fácil escribir un programa BASIC que tardase pocas horas en hallar los factores primos (difícilmente se podía suponer que hubiera factores superiores a 250000). En una semana tenía ya escrito el programa de la figura 1, que tardaba 35 minutos en hallar los dos primeros divisores del 79 y el 5651. Para no estar escrito en código máquina, era bastante rápido: Sólo hubiera tardado cerca de un año en hacer la descomposición. A las cuatro horas, desesperado, apagué mi Spectrum.

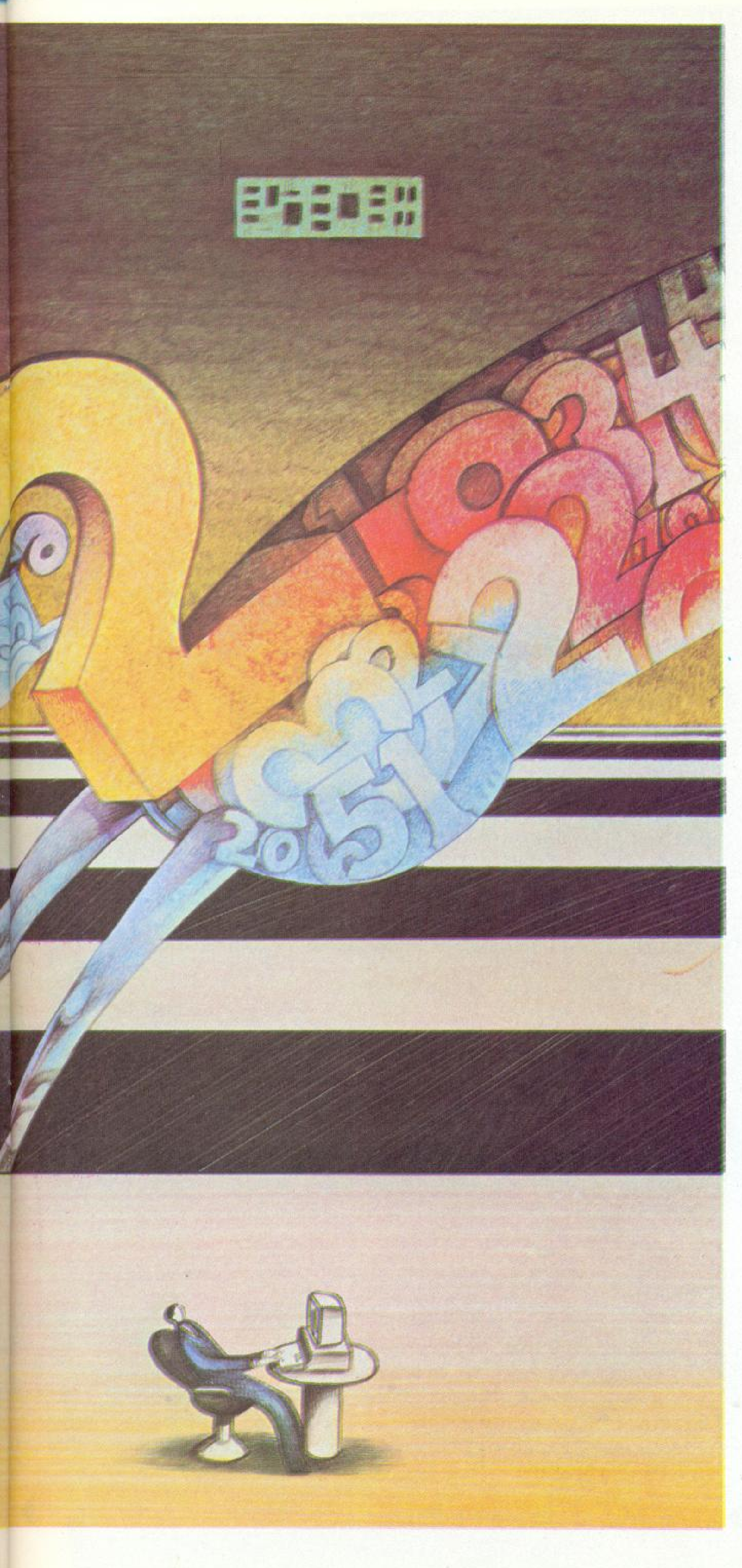
Despues de varios días intentando perfeccionar el programa sin conseguir mejoras sustanciales, me decidí por el código máquina. Primero fue una subrutina que eliminaba como posibles factores aquellos números que fueran múltiplos de los 53 primeros números primos, y que también está en el programa definitivo. Con esto hallé el tercer factor, el 327979.

Aún así no bastaba, tuve que escribir entonces en código máquina las rutinas más importantes, intentando reducir en lo posible las vueltas al BASIC. Conseguí después de varios días, un programa capaz de manejar los números en binario, con un divisor de hasta 3 bytes, y que sólo necesitaba volver al BASIC al encontrar un factor o al alcanzar la raíz cuadrada del dividendo, momento en el programa debe terminar, lógicamente. Pero tras varias horas de trabajo, el programa señalaba que no valía con tres bytes: El siguiente factor estaba por encima de 16777215 y había que introducir la posibilidad de trabajar con 4 bytes. Rápidamente me puse manos a la obra y dejé el programa como se ve en los listados de la figura 2.

Este programa dio, finalmente, los resultados esperados (79, 5651, 327979, 82347247 y 1947309701) en la noche del día 14 al 15 de enero, día en que se cerraba el plazo.

Como se ve si se intenta realizar los listados, tuve que sacrificar la estructuración del programa en favor de la velocidad; se hace, por tanto, necesario, explicar el funcionamiento del programa:

– Una vez comprobado que el 2 no es divisor del número, se coloca mediante un GOSUB 1000, el dividendo en 13 bytes a partir de la dirección 35800; el primero de estos bytes es 0, y los bytes más significativos de su raíz cuadrada en las di-



recciones 35905 y 35906. El primer divisor, que es 3, se guarda en 4 bytes a partir del 35900.

Una vez hecho esto, llama por primera vez a la subrutina en código máquina en la dirección «com».

El primer LDIR coloca la tabla de la figura 3 en la dirección 35500. Esta tabla es usada por la rutina que va desde las líneas 80 a 190 y está compuesta por 53 grupos de 2 bytes cada número (un grupo por cada número primo entre el 3 y el 251). Cada grupo está formado por un contador decreciente y por una constante, que es el número primo correspondiente.

Posteriormente, el programa calcula si el divisor es de 1, 2, 3 o 4 bytes y llama, respectivamente, a las rutinas dv1, dv2, dv3 o dv4, que realizan la división en base 2 del dividendo entre el divisor , y dan como resultado únicamente el resto. La figura 5 muestra dos ejemplos de esta división.

Lo primero que hacen estas rutinas es saltar a «setdv», donde se mira en el registro c si el divisor es múltiplo de uno de los 53 números primos. En caso afirmativo, no realiza la división y salta a «suma2». Si, por el contrario, el registro C vale 0, Coloca el dividendo en la posición 35840, sobre la que se operan y vuelve.

Luego, por otro LDIR, coloca el divisor en la posición 36000; en este momento los registros IX e IY señalan la dirección del dividendo y del divisor, respectivamente. Después del complicado proceso de la división, el registro IX se queda señalando al resto. Desde aquí el flujo del programa salta a la etiqueta «resto», que comprueba el valor del resto; si este es 0, salta a «divs», desde donde retorna el BASIC señalando que ha encontrado un divisor. Cuando esto ocurre, el BASIC lo guarda en la matriz n$ y divide el dividendo por el divisor mediante un GOSUB 4000.

Si el resto no es 0, el siguiente paso consiste en sumar dos al divisor para obtener el próximo divisor. El programa realiza entonces una última comprobación, la de si el divisor ha rebasado la raíz cuadrada, y si todavía no lo ha alcanzado cierra el ciclo en la dirección «mloop». Si lo ha rebasado ya vuelve al BASIC. En este caso, el dividendo es el último factor y el programa termina.

J. INGNACIO PEREA

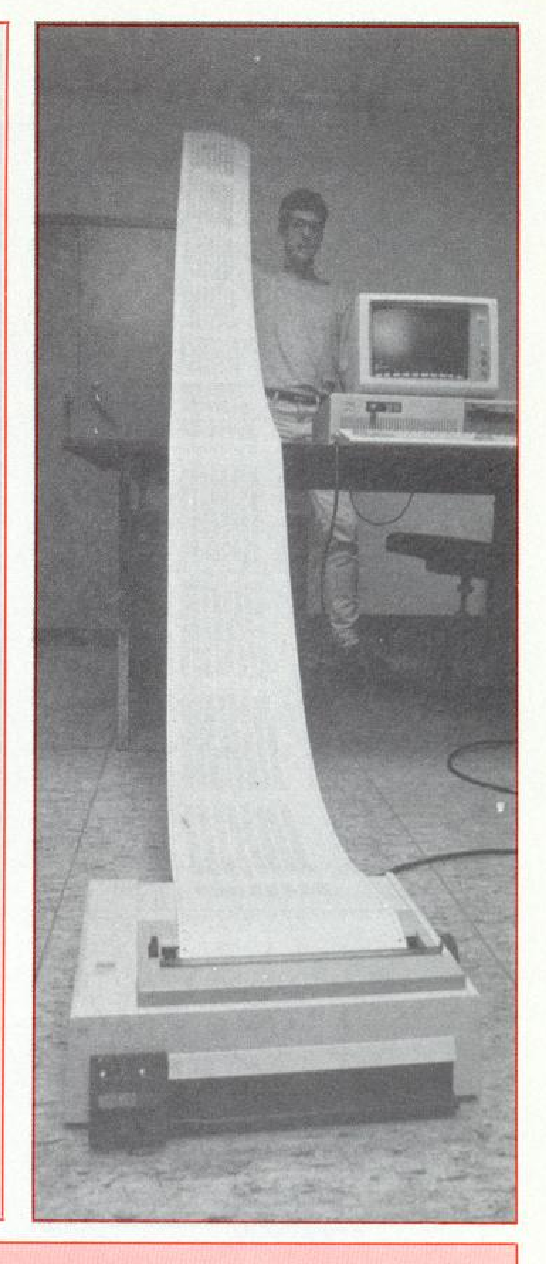
PROGRAMA 1

```
  5 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
 10 LET z$="23479159600157356175629850677"
 100 LET n=1: DIM n(15): CLS : PRINT z$
 110 LET x=2: GO SUB 500
 120 LET x=3
 130 IF x/3<>INT (x/3) AND (STR$ x)(LEN STR$ x)<>"5" AND x/7<>INT (x/7) OR x<9 THEN  GO SUB 500
 140 LET x=x+2: GO TO 130
 500 PRINT AT 21,16;x: IF x>SQR VAL z$ THEN  GO TO 8000
 600 LET j$=z$: LET k$=STR$ x: GO SUB 4000: IF VAL m$ THEN  RETURN
 700 LET n(n)=x: LET n=n+1: LET z$=l$: BEEP 1,30: PRINT AT 3*n-5,0;x''l$: GO TO 500
 990 STOP
4000 REM j$=k$ *l$ +m$
4050 IF VAL j$<VAL k$ THEN  LET l$="0": LET m$=j$: RETURN
4100 IF VAL j$<1e8 THEN  LET l$=STR$ INT (VAL j$/VAL k$): LET m$=STR$ (VAL j$-VAL k$*VAL l$): RETURN
4150 IF VAL k$>1e8 THEN  GO TO 4500
4200 LET l$="": LET j=VAL j$( TO 8)
4210 FOR i=9 TO LEN j$
4220 LET j=j*10+VAL j$(i): IF j<1e8 AND i<LEN j$ THEN  NEXT i
4225 LET l=INT (j/VAL k$): LET j=j-l*VAL k$: LET l$=l$+STR$ l
4230 NEXT i
4240 LET m$=STR$ j
4490 RETURN
4500 PRINT 4500: STOP
8000 IF z$="1" THEN  LET x=VAL z$: GO TO 700
9000 PRINT "Este programa ha tardado "'(PEEK 23672+256*PEEK 23673+65536*PEEK 23674)/50;" segundos"'"en la descomposicion."
9010 FOR f=1 TO n-1: PRINT n(f): NEXT f
9020 FOR f=1 TO 60: BEEP .01,f: BEEP .01,f/2: NEXT f
9900 STOP
9990 RUN
9999 CLEAR : SAVE "Primos" LINE 9990
```

PROGRAMA 2

```
00010        ORG 34000
00020 ;
00030 com    DI
00040        LD DE,35500
00050        LD HL,34800
00060        LD BC,106
00070        LDIR   ;tabla
00080 mloop  LD BC,13568
00090        LD HL,35500
00100 loop   DEC (HL)
00110        JR NZ,noig
00120        INC HL
00130        LD A,(HL)
00140        DEC HL
00150        LD (HL),A
00160        INC C
00170 noig   INC HL
00180        INC HL
00190        DJNZ loop
00200        JR cont
00210 rep    DI
00220        LD C,0
00230 cont   LD IY,35900
00240        LD A,(IY+0)
00250        AND A
00260        JP NZ,dv4
00270        OR (IY+1)
00280        JP NZ,dv3
00290        OR (IY+2)
00300        JR NZ,dv2
00310 dv1    CALL setd1
00320        LD HL,35903
00330        LD DE,36000
00340        LD BC,2
00350        LDIR
00360        LD IY,36000
00370        LD B,12
00380 lp1    LD A,(IX+0)
00390        CP (IY+0)
00400        JR C,seg1
00410        JR NZ,res1
00420        LD A,(IX+1)
00430        CP (IY+1)
00440        JR C,seg1
00450 res1   LD A,(IX+1)
00460        SUB (IY+1)
00470        LD (IX+1),A
00480        LD A,(IX+0)
00490        SBC A,(IY+0)
00500        LD (IX+0),A
00510 seg1   SRL (IY+0)
00520        RR (IY+1)
00530        JR NC,lp1
00540        LD (IY+2),128
00550        INC IX
00560        INC IY
00570        DJNZ lp1
00580        JP resto
00590 dv2    CALL setdv
00600        LD HL,35902
00610        LD DE,36000
00620        LD BC,3
00630        LDIR
00640        LD IY,36000
00650        LD B,11
00660 lp2    LD A,(IX+0)
00670        CP (IY+0)
00680        JR C,seg2
00690        JR NZ,res2
00700        LD A,(IX+1)
00710        CP (IY+1)
00720        JR C,seg2
00730        JR NZ,res2
00740        LD A,(IX+2)
00750        CP (IY+2)
00760        JR C,seg2
00770 res2   LD A,(IX+2)
00780        SUB (IY+2)
00790        LD (IX+2),A
00800        LD A,(IX+1)
00810        SBC A,(IY+1)
00820        LD (IX+1),A
00830        LD A,(IX+0)
00840        SBC A,(IY+0)
00850        LD (IX+0),A
00860 seg2   SRL (IY+0)
00870        RR (IY+1)
00880        RR (IY+2)
00890        JR NC,lp2
00900        LD (IY+3),128
00910        INC IX
00920        INC IY
00930        DJNZ lp2
00940        JP resto
00950 dv3    CALL setdv
00960        LD HL,35901
00970        LD DE,36000
00980        LD BC,4
00990        LDIR
01000        LD IY,36000
01010        LD B,10
01020 lp3    LD A,(IX+0)
01030        CP (IY+0)
01040        JR C,seg3
01050        JR NZ,res3
01060        LD A,(IX+1)
01070        CP (IY+1)
01080        JR C,seg3
01090        JR NZ,res3
01100        LD A,(IX+2)
01110        CP (IY+2)
01120        JR C,seg3
01130        JR NZ,res3
01140        LD A,(IX+3)
01150        CP (IY+3)
01160        JR C,seg3
01170 res3   LD A,(IX+3)
01180        SUB (IY+3)
01190        LD (IX+3),A
01200        LD A,(IX+2)
01210        SBC A,(IY+2)
01220        LD (IX+2),A
01230        LD A,(IX+1)
01240        SBC A,(IY+1)
01250        LD (IX+1),A
01260        LD A,(IX+0)
01270        SBC A,(IY+0)
01280        LD (IX+0),A
01290 seg3   SRL (IY+0)
01300        RR (IY+1)
01310        RR (IY+2)
01320        RR (IY+3)
01330        JR NC,lp3
01340        LD (IY+4),128
01350        INC IX
01360        INC IY
01370        DJNZ lp3
01380        JP resto
01390 dv4    CALL setdv
01400        LD HL,35900
01410        LD DE,36000
01420        LD BC,5
01430        LDIR
01440        LD IY,36000
01450        LD B,9
01460 lp4    LD A,(IX+0)
01470        CP (IY+0)
01480        JR C,seg4
01490        JR NZ,res4
01500        LD A,(IX+1)
```

```
         CP   (IY+1)
         JR   C,seg4
         JR   NZ,res4
         LD   A,(IX+2)
         CP   (IY+2)
         JR   C,seg4
         JR   NZ,res4
         LD   A,(IX+3)
         CP   (IY+3)
         JR   C,seg4
         JR   NZ,res4
         LD   A,(IX+4)
         CP   (IY+4)
         JR   C,seg4
res4     LD   A,(IX+4)
         SUB  (IY+4)
         LD   (IX+4),A
         LD   A,(IX+3)
         LD   A,(IX+3)
         SBC  A,(IY+3)
         LD   (IX+3),A
         LD   A,(IX+2)
         SBC  A,(IY+2)
         LD   (IX+2),A
         LD   A,(IX+1)
         SBC  A,(IY+1)
         LD   (IX+1),A
         LD   A,(IX+0)
         SBC  A,(IY+0)
         LD   (IX+0),A
seg4     SRL  (IY+0)
         RR   (IY+1)
         RR   (IY+2)
         RR   (IY+3)
         RR   (IY+4)
         JR   NC,lp4
         LD   (IY+5),128
         INC  IX
         INC  IY
         DJNZ lp4
resto    LD   A,(IX+0)
         OR   (IX+1)
         OR   (IX+2)
         OR   (IX+3)
         JR   Z,divis; Vuelve
         LD   BC,254 ; si CS,
         IN   A,(C)  ; A,Q,1,
         RRA         ; O,P,ENT
         JR   C,suma2; o SPC
         LD   A,2    ; estan
         JR   volv   ;pulsados
tecla    DI
suma2    LD   A,2        ;Suma 2
         LD   IY,35900 ;al di-
         ADD  A,(IY+3);visor
         LD   (IY+3),A
         LD   A,(IY+2)
         ADC  A,0
         LD   (IY+2),A
         LD   A,(IY+1)
         ADC  A,0
         LD   (IY+1),A
         LD   A,(IY+0)
         ADC  A,0
         LD   (IY+0),A
         LD   H,A        ;(IY+0)
         LD   L,(IY+1)
         LD   D,(IY+5)
         LD   E,(IY+6)
         SBC  HL,DE
         JP   C,mloop;limite
         XOR  A        ;rebasado
         JR   volv
divis    LD   A,1
         JR   volv
setdv    LD   A,C        ;Numero
         AND  A          ;primo?
         JR   Z,setd1
         POP  HL;dir.para RET
         JR   suma2
setd1    LD   HL,35800
         LD   DE,35840
         LD   BC,16
         LDIR       ;dividendo
         LD   IX,35840
         RET
volv     LD   (23296),A
         EI
         RET     ;al BASIC
Final    DEFB 0,0,0,0,0
```



PROGRAMA 3

```
   5 GO TO 20
  10 CLEAR 26999: LOAD ""CODE
  20 DIM n$(21,12): LET n=1
 100 CLS : LET w$="2347915960015
7356175629850677": INPUT  1986
 Perea Software"'''"Numero:"'"(P
resiona ENTER para el numero"'(w
$);")"'p$: PRINT AT 21,16;"BASIC
"
 105 IF p$="" THEN  LET p$=w$
 107 LET y$=p$: PRINT AT 21,16;"
BASIC";AT 0,0;y$
 110 LET a=VAL y$(LEN y$): IF a<
>2*INT (a/2) THEN  GO TO 180
 120 LET x=2: LET z$=y$: GO SUB
4000: LET y$=l$: LET n$(n)="2":
LET n=n+1: PRINT "2 ";y$: GO TO
110: REM division por 2
 180 POKE 35813,0: POKE 35814,0:
 POKE 35815,0: GO SUB 1000: POKE
 35900,0: POKE 35901,0: POKE 359
02,0: POKE 35903,3: POKE 35904,0
: REM primer divisor=3
 190 BEEP .1,0: PRINT AT 21,16;"
 C/M ": LET us=USR 34000
 200 PRINT AT 21,16;"BASIC": LET
 us=PEEK 23296
 205 LET x=16777216*PEEK 35900+6
5536*PEEK 35901+256*PEEK 35902+P
EEK 35903
 210 IF us=0 THEN  BEEP .1,-20:
GO TO 8000
 215 IF us=2 THEN  BEEP .01,60:
PRINT AT 21,0;x: PRINT AT 21,16;
" C/M ": RANDOMIZE USR 34534: GO
 TO 200
 217 REM  IF us=1 THEN
 220 BEEP .1,40: LET z$=y$: GO S
UB 4000: LET n$(n)=STR$ x: PRINT
 AT n,0;x;" ";l$: LET n=n+1: LET
 y$=l$
 222 IF x>SQR VAL y$ THEN  GO TO
 8000: REM No buscar mas
 225 GO SUB 1000: BEEP .1,0: PRI
NT AT 21,16;" C/M ": RANDOMIZE U
SR 34032: GO TO 200
1000 REM  Operaciones previas
          para cada dividendo
1005 LET x=256: LET z$=y$: FOR f
=35812 TO 35800 STEP -1: GO SUB
4000: POKE f,j: LET z$=l$: NEXT
f
1010 LET a=SQR VAL y$/65536+1: R
EM      +1 por si acaso
1015 IF a>65535 THEN  LET a=6553
5: REM limite aproximado para
        seguir calculando
1020 POKE 35905,INT (a/256): POK
E 35906,a-256*INT (a/256): RETUR
N
4000 REM Division entera  .-
        z$ ▌ x
        ──────
        \j/  l$
4100 IF VAL z$<1e8 THEN  GO TO 4
500
4200 LET l$="": LET j=VAL z$( TO
 7)
4210 FOR i=8 TO LEN z$
4220 LET j=j*10+VAL z$(i)
4225 LET l=INT (j/x): LET j=j-l*
x: LET l$=l$+STR$ l
4230 NEXT i
4490 RETURN
4500 LET l$=STR$ INT (VAL z$/x):
 LET j=VAL z$-x*VAL l$: RETURN
7900 REM Encontrados todos los
          factores primos
8000 IF y$<>"1" THEN  LET n$(n)=
y$: LET y$="1": LET n=n+1
9000 BEEP .5,30: CLS : PRINT "Lo
s factores primos son:": FOR f=1
 TO n-1: PRINT n$(f): NEXT f
9300 REM Reloj
9400 POKE 23672,0: POKE 23673,0:
 POKE 23674,0
9500 POKE 23624,184: PRINT )0;AT
 0,0;"  Terminado hace ";INT ((6
5535*PEEK 23674+256*PEEK 23673+P
EEK 23672)/50);" segundos.","
    Presiona ENTER",: IF INKEY$
<>CHR$ 13 THEN  GO TO 9500
9998 BORDER 7: STOP
9999 CLEAR : SAVE "PROGRAMA" LIN
E 10: SAVE "pr c\m"CODE 34000,95
0: REM Programa+Bytes
```

FIGURA 4

valor del divisor	contador => 3	valor de cada contador						
comienzo	1	2	3	5	6	8	9	-----------
3	0-3	1	2	4	5	7	8	-> multiplo de 3 primo
5	2	0-5	1	3	4	6	7	-> multiplo de 5 primo
7	1	4	0-7	2	3	5	6	-> multiplo de 7 primo
9	0-3	3	6	1	2	4	5	-> multiplo de 3
11	2	2	5	0-11	1	3	4	-> multiplo de 11 primo
13	1	1	4	12	0-13	2	3	-> multiplo de 13 primo
15	0-3	0-5	3	9	12	1	2	-> multiplo de 3 y 5
17	2	4	2	8	11	0-7	1	-> multiplo de 17 primo
19	1	3	1	7	10	16	0-19	-> multiplo de 19 primo
21	0-3	2	0-7	6	9	15	18	-> multiplo de 3 y 7
23	2	1	6	5	8	14	17	-> multiplo de 23 primo
-	-	-	-	-	-	-	-	- - - - - - - - -
-	-	-	-	-	-	-	-	- - - - - - - - -
-	-	-	-	-	-	-	-	- - - - - - - - -
257	2	4	1	9	8	16	14	=> primo
259	1	3	0-7	8	7	15	13	
ect	-	-	-	-	-	-	-	

FIGURA 5

```
a) 39:3 -> 100111:11

     100111 -->  100111
   - 110000      -11000
     ------      ------
                 001111   --> 1111
                            - 1100
                              ----
                               011  -->  011  -->  11
                                        -110      -11
                                         ---       --
                                                    0 --> resto 0 -->divisible

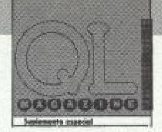
b) 47:3 -> 101111: 11

    101111   --->  101111
   -110000         -11000
    ------         ------
                   10111    ---> 10111
                                 -1100
                                 -----
                                  1011 --->  1011
                                             -110
                                             ----
                                              101 ----> 101
                                                         11
                                                        ---
                                                         10  ---->resto 2; no
                                                                     divisible
```



Quién es quién en el mundo del QL

Los BUGS del sistema operativo



Quien es quien en el mundo del QL

Una de las mayores preocupaciones de los usuarios del QL es la escasez de software y hardware en España. Cada vez son más quienes recurren al mercado británico, dado el desinterés de Investrónica. Por ello, ofrecemos una lista de las empresas británicas que trabajan para el QL, especificando los productos que comercializan.

PROSPERO SOFTWARE LTD.
190 CASTLENAU
LODON SW13 9D4

Esta firma desarrolla actualmente software para el QL bajo el sistema operativo QDOS. Prospero es conocida por sus programas para CP/M-80 y el PRO-PASCAL-ISO para IBM PC, primer compilador de Pascal para microordenador que superó las pruebas de homologación del I.S.O.

DIGITAL PRECISION
222 THE AVENUE
LONDON E4 9SE

Prácticamente desconocida hasta ahora, Digital Precision ha sido lanzada al estrellato por el SUPERCHARGE SUPERBASIC COMPILER (Compilador de SUPERBASIC). Otros productos para QL desarrollados por esta empresa son:

Super Forth
Super Sprite Generator
Super Monitor Dissasembler

QJUMP
25 KING STREET
RAMPTON
CAMBRIDGE CB4 4QD

QJUMP fue fundada por Tony Tebby, creador del sistema operativo del QL. Entre al software desarrollado por esta empresa se encuentra el QL TOOLKIT, extraordinario programa completo al sistema operativo. También comercializan los siguientes productos:

QL Toolkit II (ROM)
QL Monitor Debugger
Eprom Programmer
QFLP Upgrade ROM

QJUMP ha escrito el firmware de la mayoría de los controladores de floppy para QL. El firmware de Tony Tebby permite leer, escribir y formatear discos en sectores de 128 bytes a 1024 bytes, lo que cubre la mayor parte de los formatos utilizados. Puede acceder al disco como si fuese un sólo fichero de éste, lo que permite el lujo de tener un floppy con discos formato PC DOS, otro QDOS y otro UNIX, por ejemplo. Y con rutinas adecuadas se podría trabajar con los tres formatos a la vez. Con las rutinas adecuadas, el QL podría ser una solución a la conversión de formatos.

CENTRONICS

Este fabricante de impresoras, que ha dado su nombre al interface paralelo para la conexión de estas, ha lanzado una impresora con el juego de caracteres del QL, terminando con los problemas de *translate*.

COMPUTER ONE
SCIENCE PARK
MILTON ROAD
CAMBRIDGE

Computer One es una empresa muy conocida en el mundo del QL. Dispone de los siguientes programas:

QL Typing Tutor
QL Assembler
QL Forth
QL Pascal

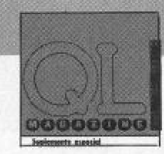
El Pascal es inferior al de METACOMCO, pero a cambio deja más memoria libre para el usuario.

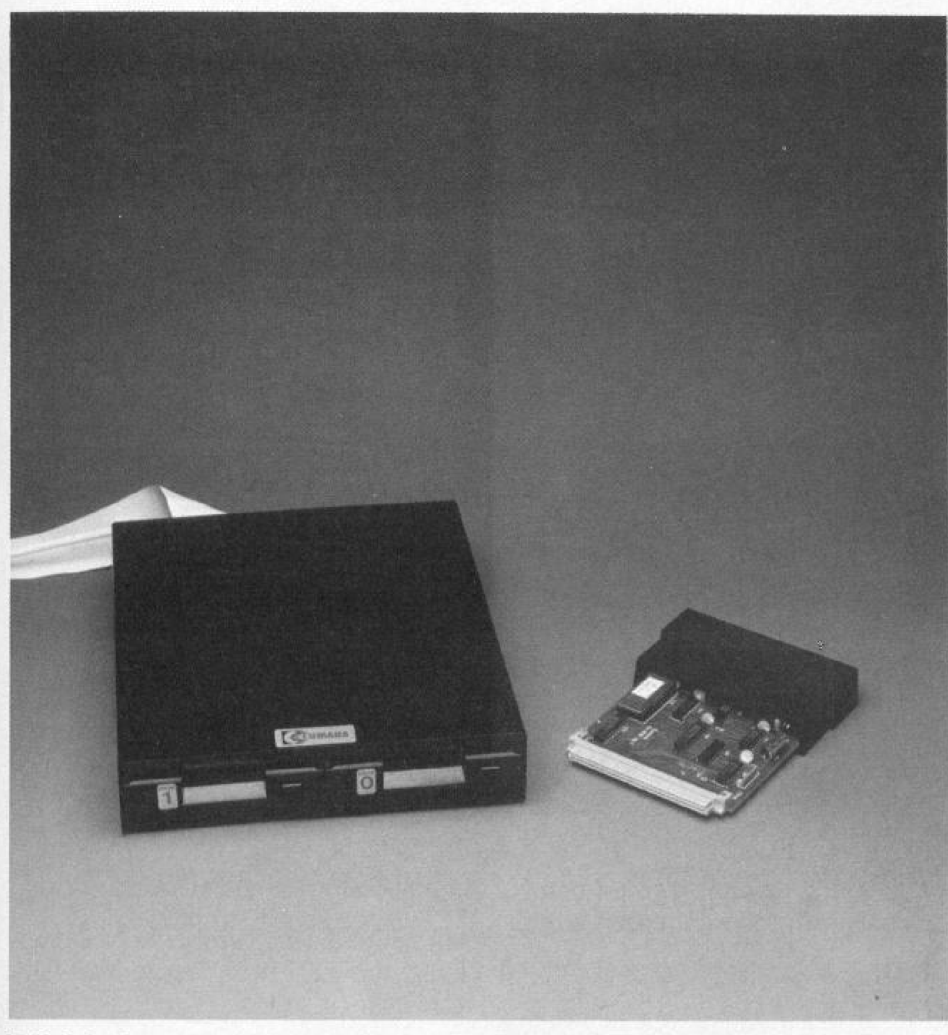
GST COMPUTER SYSTEM LTD.
91 HIGH STREET
LONG STANTON
CAMBRIDGE

De GST destacan el ASSEMBLER y el LINKER, programas adoptados oficialmente por Sinclair.

QUEST INTERNATIONAL LTD.
TEL.: 04215 66488

Destaca la implementación del sistema operativo CPM/68K, origi-





Unidad de discos y controlador de Cumana.

nal de DIGITAL RESEARCH, disponible desde septiembre de 1984. También distribuye paquetes de aplicaciones y juegos, tanto para CPM como para QDOS.

En cuanto a hardware, cuenta con ampliaciones de memoria de 64 a 512 K, unidades de disco de 5"1/4 y un disco Winchester de 7,5 MB.

MICRO APL.
TEL.: 01 622 0395

Esta firma, especializada en intérpretes de APL, distribuye al APL del QL. Este potente lenguaje estaba disponible hasta ahora únicamente para grandes ordenadores.

TANDATA MARKETING LTD.
ALBERROAD NORTH
MALVERN
WORCS WR14 2TL

Distribuye el QCOM, interface de comunicaciones, el QCALL, módulo de llamada, marcado y respuesta automática y el QMOD, modem. Juntos constituyen un sistema completo de comunicaciones para el QL, permitiendo la conexión a bases de datos de grandes ordenadores.

DATA MANAGEMENT
CLARK HOUSE
THE VILLAGE
HAXBY
YORK Y03 8HU

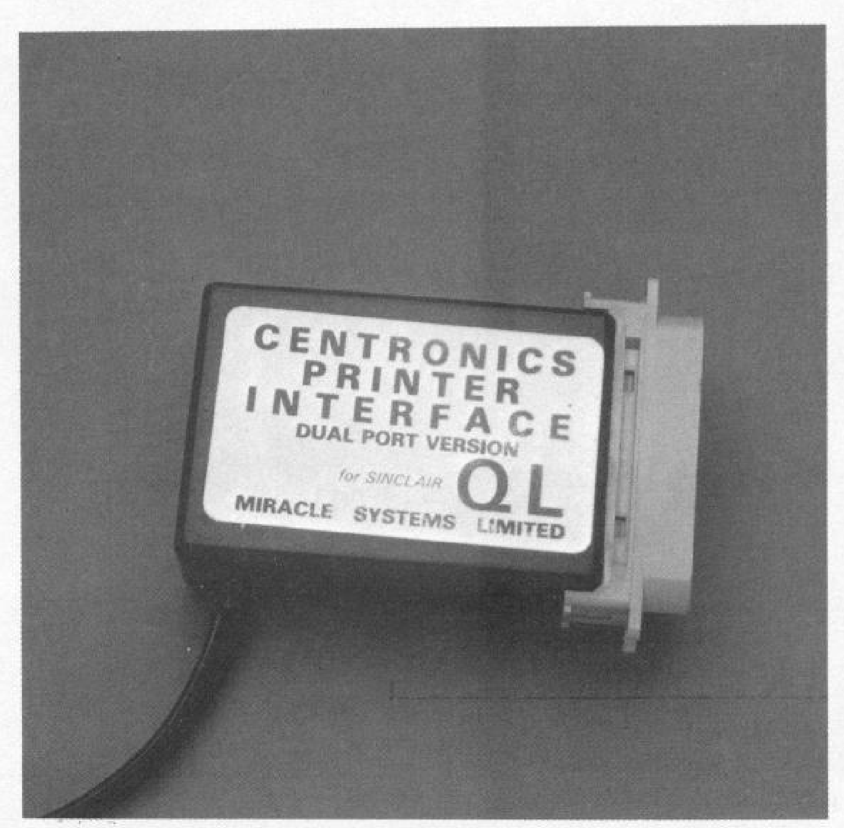
Otra firma que dedica sus esfuerzos al terreno de las comunicaciones con emuladores múltiples de terminal, software de comunicaciones y modems.

MICROPERIPHERAL LTD.
TEL.: (0256) 473232

Fabricante de los discos oficiales de Sinclair. Estos discos han sido criticados por tener un firmware inferior al de otras unidades.

COMPUTAMATE
TEL.: 0768 811711

Una de las primeras casas en fabricar discos de 3.5 y 51/4 para el QL. El software es de QJUMP. También realiza consolas de expansión.



Adaptador serie-Centronins de Miracle Systems.

PCML
ROYAL MILLS
ESHER, SURREY KT10, 8AS

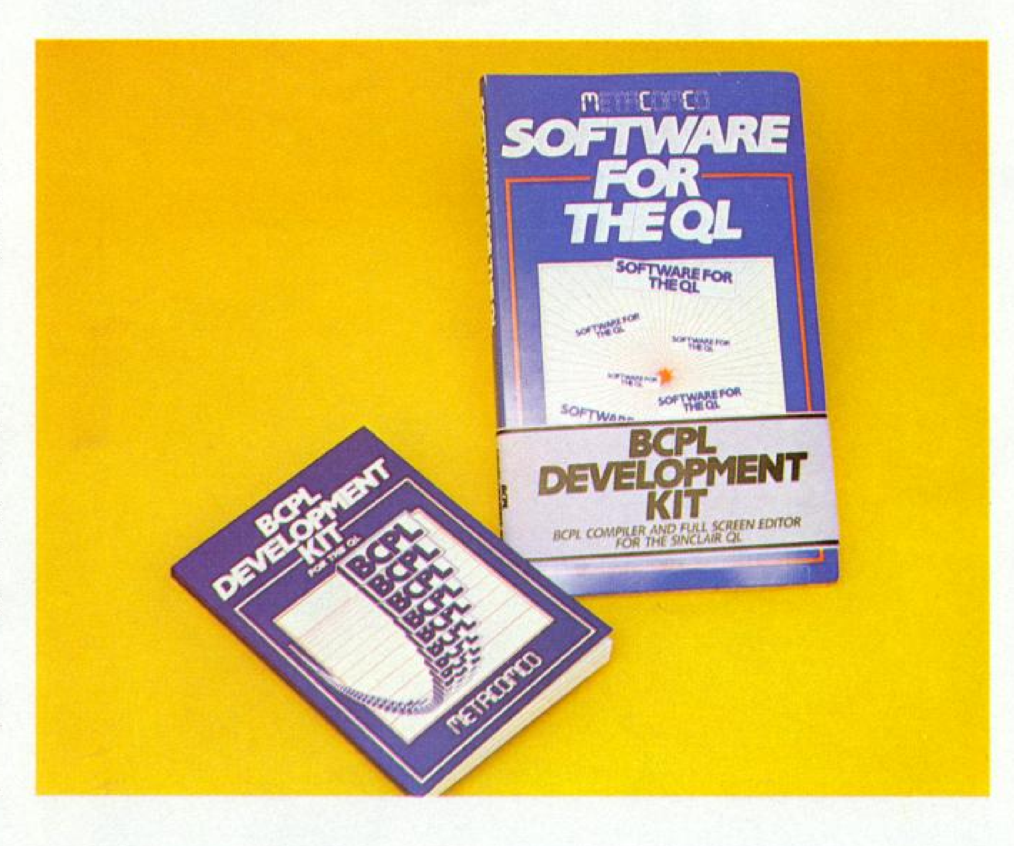
Dispone de expansiones de memoria, interfaces de floppy con RAM adiccional y de una tarjeta con un procesador Z-80, lo que permite la posibilidad de utilizar CPM 80.

TDI, LTD.
BRISTOL

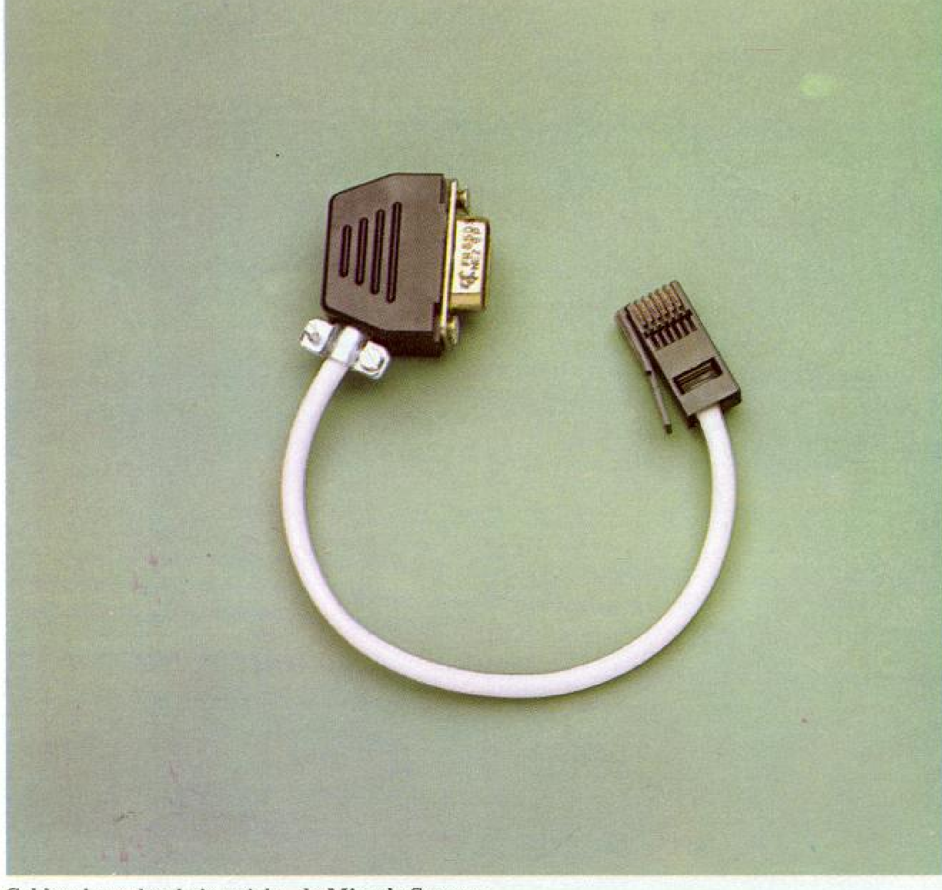
TDI ha desarrollado el sistema operativo P-SYSTEM o UCSD-PL, creado en la Universidad de California—San Diego. Para este sistema se han lanzado inicialmente los populares UCSD-PASCAL y FORTRAN-ANSI 77, así como diversas aplicaciones.

MIRACLE SYSTEMS LIMITED
UNIT 37a, AVONDALED WORKSHOPS
WOODLAND WAY, KINGSWOOD
BRISTOL BS15 1QL

Produce ampliaciones de memoria de 256 K y 512 K, con un conector que permite la conexión simultánea del disco. También comercializa convertidores serie paralelo Centronics.



BCPL, desarrollado por Metacomco.



Cable adaptador de joysticks, de Miracle System.

CAMBRIDGE SYSTEMS TECHNOLOGY (CST)
30 REGENT STREET
CAMBRIDGE CB2 1DB

Entre sus productos destaca el superprofesional interface IEEE-488, creada por el Instituto de Ingenieros Eléctricos y Electrónicos y adoptado como standard mundial por la Comisión Electrototécnica Internacional (IEC).

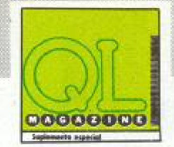
METACOMCO
26 PORTLAND SQUARE
BRISTOL BS2 8RZ

Junto con Computer One, fue una de las primeras empresas en producir software para el QL.

Cuenta con los siguientes programas:

Assembler
BCPL
LISP
Pascal-ISO
Lattice-C

JOSE M. GUZMAN



LOS BUGS DEL SISTEMA OPERATIVO

AUNQUE el QL tiene un número muy pequeño de bugs, es conveniente conocerles, para saber evitarlos.

En primer lugar vamos a citar los bugs comprobados en la versión española, que son tres únicamente, y dos de ellos en el WHEN ERRor.

1 La instrucción GOSUB dentro de un bucle corto FOR o REPeat actúa con fin del bucle.

Forma de evitarlo: utilizar la forma larga, o utilizar las más racionales procedures.

2 Hay un indicador de error que cuelga el QL.

Forma de evitarlo: no utilizar indicadores, es mejor utilizar el ERNUM y una variable. Ejemplo INDIC=ABS(ERNUM) y luego if o select con la variable INDIC. Los valores de INDIC están en el manual en ERRORES, la función ERNUM los da en negativo, de ahí el ABS.

3 Cuando está el WHEN ERRor activado, y al llamar a una función con una expresión en su argumento se produce el error de desbordamiento, al calcular ésta se cuelga el QL.



Forma de evitarlo: utilizar una variable intermedia de forma que no se llame a las funciones con expresiones dentro del paréntesis.

d=sqrt(3*X/12−10) puede colgar el QL con WHEN ERRor.

d=3*x/12−10:d=sqrt(d) no lo cuelga nunca.

Creo que por esos dos pequeños bugs, no debemos de abandonar el uso del estupendo WHEN ERRor.

Hay detalles que no son bugs, sino actuaciones racionales de los diseñadores del QL:

El mensaje de "error en expresión" o de "nombre incorrecto" ocurre siempre que se encuentra con una variable no inicializada, procedures o funciones no definidas, etc. Por cierto, el ANSI Basic exige este mensaje de error para variables no inicializadas.

No admite el dimensionamiento implícito, al igual que el ANSI BASIC, hay que definir siempre las dimensiones de las matrices.

En un WHEN ERRor no se admiten errores, para evitar bucles de autotratamiento de errores, y pone el mensaje de:
"durante una ejecución WHEN"
para volver a la normalidad pulsar directo desde el teclado END WHEN.

El «break» no es atrapado por WHEN ERRor, para facilitar la corrección de programas que lo utilicen.

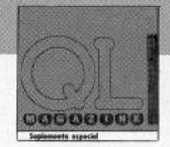
Debido a que se trazan por «stipples» bloques de cuatro puntos, al trazar puntos traza dos en vez de uno, es fácil definirse una función punto con BLOCK, o solicitar la rutina de Investrónica para un solo punto.

En procedures y subrutinas no se admite MRUN y MERGE para evitar que se pueda desordenar la pila de retornos de subrutinas.

RENUM es peligroso desde un programa, puede liar al intérprete si cambia el número de línea en que se solicita, también DLINE es peligroso desde programa, si borra la línea que se está ejecutando, los dos no se admiten desde procedures.

Hay una combinación extraña de teclas que «cuelga» al QL sino se tiene un software adecuado, es debido a que genera una interrupción nivel 7. Sirve pra que los programadores puedan intentar levantar «cuelgues» de programas en código máquina. Programas monitores especiales aprovechan esta *fantástica facilidad del QDOS.*

Para evitar lios, la radiodifusión



de red local funciona sin comprobación, sólo para cortos mensajes, debe de utilizarse sólo como canal de órdenes, y disponer de la propia comprobación por programa de los usuarios. Viene documentado en el Manual de usuario, Máxima longitud de mensaje en Radiodifusión por red local 255 bytes. Que nadie hable de Bugs si falla en mensajes largos. Para mensajes largos utilizad la red normal tal y como indica el manual.

Por último, antes de hablar de un nuevo bug, consultar, es muy fácil que sea error de programación.

Veamos ahora la «TERRIBLE» lista de bugs, algunos muy oscuros,- de las versiones anteriores. No existen en la versión española, excepto algunos no comprobados. Aquellos que tengan QLs antiguos, que cambien a las nuevas ROMS.

1 Una muy molesta para los programadores en C.M.: el QL se colgaba al utilizar CALL desde un programa de SuperBasic que ocupase más de 32K. No se ha comprobado, pero Sinclair daba una rutina que lo corregía. Habitualmente se cargan extensiones en C.M. en el programa BOOT, que es corto. No es bueno cargar en el programa principal las extensiones, si se para y se vuelve a lanzar, va rellenando cada vez la memoria de rutinas en C.M. y se puede forzar un fuera de la memoria por mala programación.

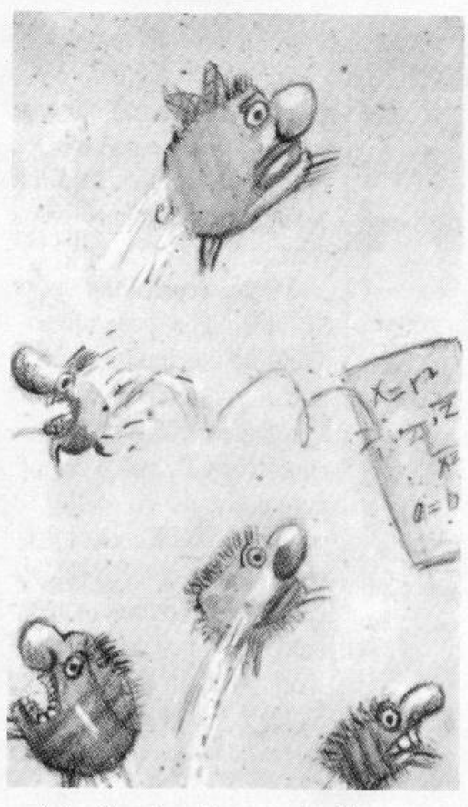
2 No reconocía más que una ROM de periféricos. Aunque no se ha comprobado experimentalmente, no hay errores en las rutinas de chequeo de ROM de periféricos, situadas entre $4A62 y $4AC2. Sinclair también proporcionaba una rutina de corrección.

3 List desde un programa daba "no implementado".

4 Si con DLINE se borra la última procedure de un programa, se llama a la procedure desde el teclado, y después se hace clear, se colgaba el QL.

5 Si se coloca una expresión entre paréntesis en los data, ignoraba el resto de la línea.

El QL admite expresiones en los DATA, y si se colocan las constantes alfanuméricas sin comillas o



apóstrofes da el correcto "error en expresión" al no encontrar la variable.

6 dir mdv8__ y después dir mdv2__ no funcionaba bien.

7 Al editar una procedure abortada, a veces editaba una línea cualquiera. Se corregía con un simple list u otra cosa y después editar.

El mensaje "PROC/FN limpiado" es correcto, siempre que se edite, o se dé un comando directo distinto de CONTINUE O RETRY, y el programa se haya parado en medio de una procedure o función, si no se hiciese así, se desordenaría la pila de retornos.

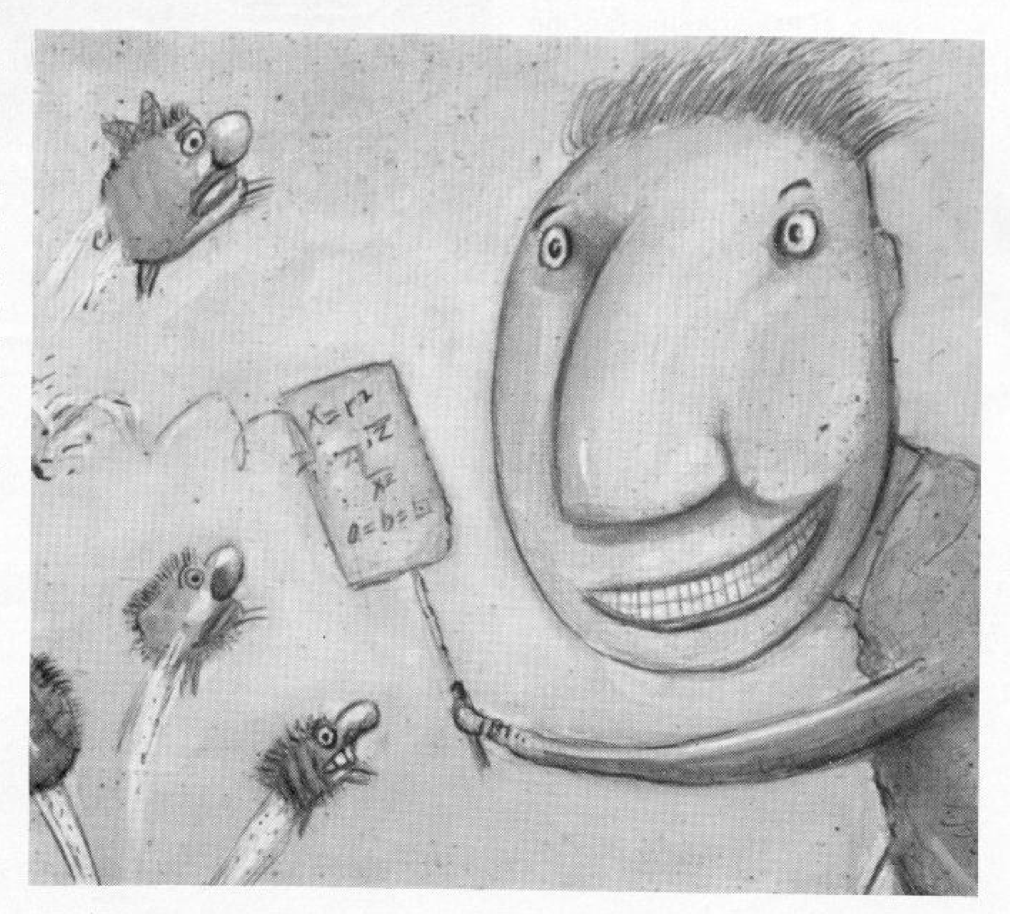
8 Cuando se declaraban más de 9 nombres de variables locales, originaban problemas.

9 No era conveniente hacer troceado de matrices de cadenas, era mejor utilizar una variable intermedia.

10 En la ROM JS, de todos los argumentos de la función, sólo admitía el SELECT con el último.

Además hay que destacar que los bugs son en su mayoría del intérprete y no del QDOS, por cierto sería genial una rutina que aprovechase la interrupción de nivel 7, para «levantar» los cuelgues del intérprete. Aunque lo bueno es compilar.

José M. Guzmán





SISTEMAS OPERATIVOS MULTITAREA

DESDE el lanzamiento del QL se habla mucho de la multitarea y del mítico UNIX. Vamos a explicar un poco la historia de los sistemas operativos multitarea.

Cuando la Informática estaba empezando, y los equipos eran muy costosos, una máquina con las posibilidades de un Spectrum costaba millones de pesetas y a nadie se le ocurría poder tenerla para él solo.

El sistema que se utilizaba para aprovechar los costosos equipos era el *batch,* o procesado por lotes. Los usuarios del ordenador depositaban sus programas y datos en el Centro de Cálculo, los empleados de centro introducían los programas y datos y recogían los resultados, luego el usuario recibía los resultados.

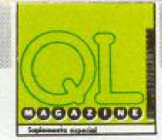
Naturalmente el menor error del programa y de los datos, obligaba a repetir el proceso, con la consiguiente pérdida de tiempo. El desarrollo de programas era laboriosísimo.

Como era económicamente imposible el que un usuario pudiese tener un ordenador para él sólo, en las universidades los programadores, cansados de entregar un programa, recogerlo al día siguiente con errores, volverlo a entregar, etc., tuvieron la genial idea de que, dado que el ordenador es mucho más rápido que el usuario, si se le hacía dedicar una pequeña fracción de su tiempo a cada usuario, a este le parecería tenerlo sólo para él, pero como el ordenador es rápido, podría atender a la vez a muchos usuarios.

Esta genial idea es el sistema de multitarea por partición de tiempo, «time-slicing», el más utilizado hoy en día.

Naturalmente, desde entonces han progresado los sistemas operativos multitarea, añadiéndose más funciones, pero la básica es ésta, la partición del tiempo, y vamos a dar una idea de cómo se realiza.

El corazón de la multitarea es el repartidos de tiempos de ejecución, el *sheduler,* es una rutina que se llama por las interrupciones hardware cada cierto tiempo, y decide qué programa va a ser ejecutado, salvando los valores necesarios para poder volver a comenzar el progrma anterior, y colocando los nuevos del pro-



grama a ejecutar.

Para aquellos que quieran examinar con un desensamblador el *scheduler* del QL, ofrecemos una somera indicación de las direcciones (hexadecimales) de las rutinas más importantes:

— $352 entrada de todo tipo de interrupciones nivel 2, comprueba si son de *microdrives,* disco, etc.

— $90A entrada de interrupciones de temporización y *scheduler.*

— $936 Comprobación de tareas de altísima prioridad (modo sistema). Si se está en ellas no se ejecuta el *scheduler.*

— $940 a $966 el *scheduler* propiamente dicho.

— $9DE a $A14 la rutina que salva todos los valores y datos necesarios para volver a reanudar la ejecución del programa suspendido.

— $A16 a $A80 la rutina que elige el programa que le corresponde ejecutarse.

— $A82 a $AA6 la rutina que coloca los valores necesarios para reanudar la ejecución de la tarea.

— $AA8 a $AC8 una interesante rutina de llamada a las listas encadenadas de tratamiento de interrupciones.

Las direcciones indicadas son las de la ROM española MGE, en otras ROM pueden cambiar.

El *scheduler* es el corazón del sistema operativo multitarea, pero un sistema multitarea necesita mucho más que eso. El hecho de que estén ejecutándose varios programas a la vez puede dar lugar a muchos problemas, por ejemplo, tomado del libro de Dickens, es fácil imaginarse un listado en que varios programas escribiesen por impresora a la vez.

El sistema operativo gestiona la utilización de los recursos del ordenador, evitando, en lo posible, que unos programas «machaquen» a otros, controla la utilización de ficheros, *interfaces,* etc. También, desde el Multics, se incorporan sistemas de comunicación intertareas, llamados *pipes,* para facilitar el uso del equipo. Los *pipes* están disponibles en el QDOS, pero no en SuperBASIC. Hay rutinas de Qunata que permiten utilizar las *pipes* desde el SuperBASIC.

Por otro lado, en los modernos sistemas multitarea, cuando un programa está esperando la realización de una operación de entrada/salida o el envío de la comunicación de un dato por otro programa, este programa se detiene para evitar que malgaste el tiempo del procesador.

Como vemos son muy complejas y variadas las labores que debe realizar un sistema operativo multitarea, por lo que se suelen implementar en ordenadores potentes. Además, al ser complejos, su desarrollo es costoso, por lo que normalmente sólo se utilizan máquinas caras. Pero tio Clive nos puso, como siempre, lo inasequible a precios razonables.

Naturalmente sistemas operativos multitarea hay muchos, empezando por el mítico UNIX y sus variantes, siguiendo con el PICK, OASIS, los Concurrent CPM 86, 286 y 68K, el Tripos del Amiga, el 68KOS y nuestro QDOS.

Y si hablamos de miniordenadores y grandes Ordenadores *(mainframes),* hoy en día no se concibe un sistema operativo para ellos sin ser multitarea, simplemente por lo menos para el *spool* de impresora (esto es imprimir en ficheros de disco, y un programa en multitarea se encarga de imprimir mientras el ordenador sigue trabajando). Realmente sería absurdo tener estos equipos parados esperandod el acabar de imprimir.

Hay muchas variantes de sistemas operativos multitarea, de tiempo real, multiusuario, etc. Pero vamos a ir empezando poco a poco.

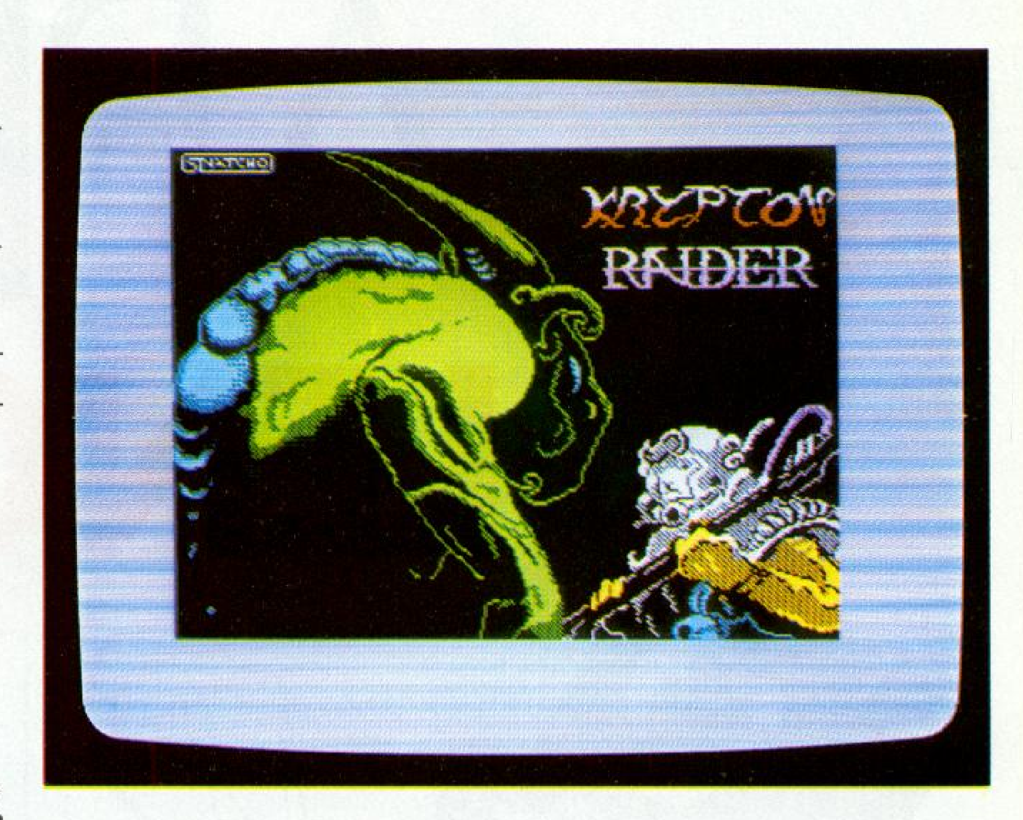
Para aquellos que quieran introducirse más a fondo en el QDOS, el libro recomendado es la **Guía del Usuario Avanzado del QL** por **Adrian Dickens,** publicado por **RAMA.**

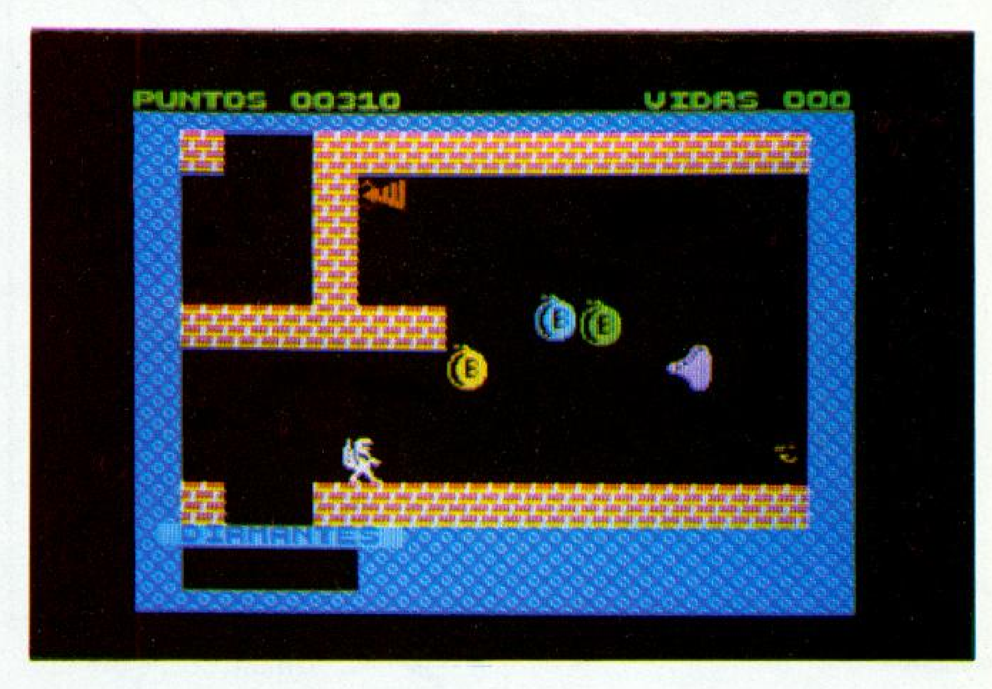
José M. Guzmán

KRIPTON RAIDERS

FUTURE STARS

SPECTRUM 48 K





Siguiendo una línea que parece estar teniendo éxito en el mercado de los juegos para Spectrum, Dinamic ha dado a luz esta nueva firma, cuyo cometido es ofrecer al sufrido consumidor nuevos títulos al precio de 999 ptas., a la vez que da a programadores nóveles una oportunidad para triunfar.

El objetivo de la aventura es, tras localizar y recoger los cuatro diamantes que se hallan diseminados al azar por un intrincado laberinto en dos dimensiones, alcanzar la nave que nos permita huir de allí. En cada pantalla que visitemos nos encontraremos cierto número de extrañas criaturas que harán difícil la tarea, por lo que constantemente tendremos que hacer uso de nuestro láser. Sin duda el mayor atractivo de este programa reside, precisamente, en la cantidad de disparos que podemos llegar a hacer en cada partida, acabando con docenas de enemigos en muy poco tiempo.

La extructura del juego nos recuerda bastante al viejo Jet-Pac de Ultimate, pero el tamaño del protagonista es menor, mientras que el mapa es más extenso. El movimiento de los sprites, sin ser malo, deja pequeños detalles sin acabar, especialmente cuando se sobreponen varios de ellos o cuando el astronauta es destruido.

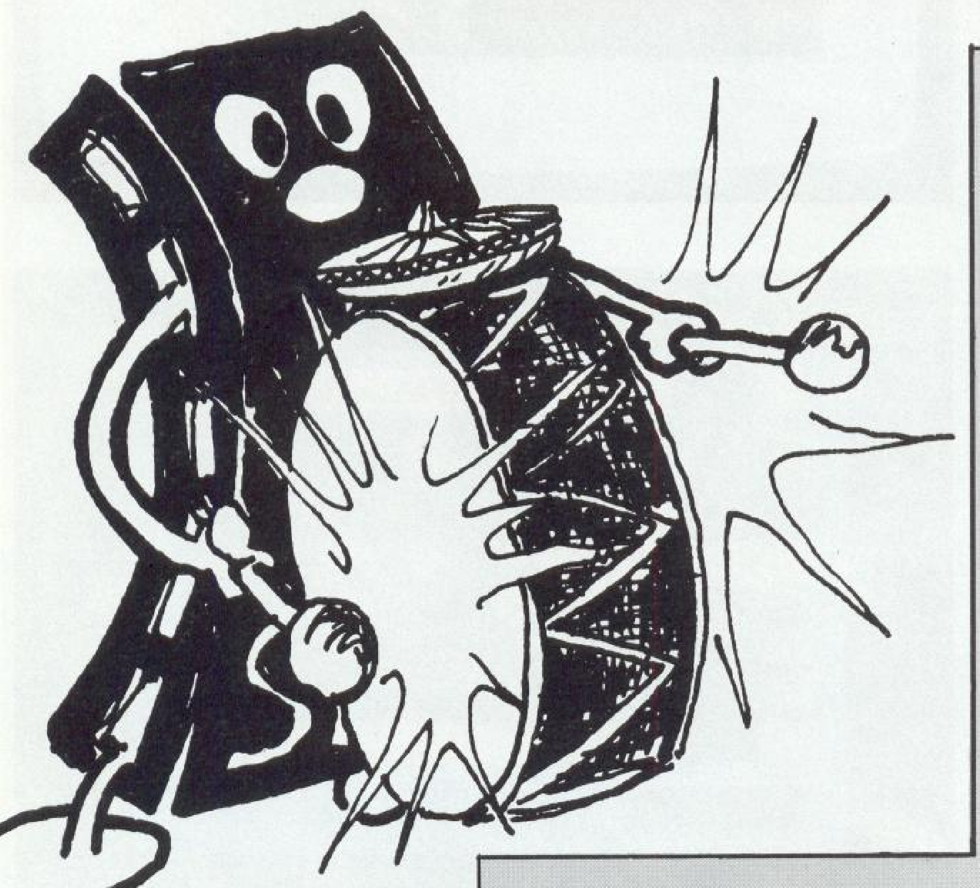
El nivel de dificultad es bastante alto, como viene siendo norma en las últimas creaciones de todas las casas. Parece ser que quien decide gastar su dinero en la compra de un programa de este tipo desea que, al menos, le dure unas semanas. Los gráficos no aportan nada nuevo, pero cumplen perfectamente con su papel.

Otro tanto hay que decir de los detalles sonoros y de presentación, que hacen que el juego sea completo en sus partes más esenciales.

Se trata, en definitiva, de un buen programa si lo calibramos teniendo en cuenta el precio al que se comercializa. Muy aconsejable para los amantes de los juegos de acción.

APREN

LENGUAJ



Cuando se pensó por primera vez en la creación de un microprocesador «hijo» del popular Intel 8080 y se empezó a calibrar cuáles serían las modificaciones más interesantes que se le podían hacer, surgieron, por un lado, voces que clamaban por una ampliación del número de registros (de ahí surgió el juego alternativo y los índices IX e IY), y, por otra parte, las de quienes creían necesaria la implementación de nuevas instrucciones que aumentaran la potencia y la versatilidad del nuevo procesador y lo hicieran más cómodo al programador. Entre estas últimas instrucciones destacan las que permiten trabajar con bits independientes en cualquier registro o posición de memoria, una de tantas virtudes que hacen al Z-80 el rey de los 8 bits.

La instrucción BIT permite que conozcamos el estado lógico de un bit cualquiera en un registro o una posición de memoria que direccionemos con HL, IX o IY. El valor del bit que nos interesa es copiado en la bandera de cero (Z), por lo que si, por ejemplo, queremos llamar a una determinada rutina, sólo en caso de que el bit 3 del registro D valga 1, podemos hacer lo siguientes:

```
BIT     3,D
CALL    Z,RUTINA
```

Como puede verse, esto resulta de especial interés cuando registros o posiciones de memoria son usados como grupos de banderas, al estilo de las variables del sistema FLAGS, FLAGS2, TVFLAGS, etc. La principal ventaja es que no es necesario modificar el valor de ningún registro como en el uso de máscaras

NDIENDO Capítulo 9

JE MAQUINA

con las instrucciones lógicas, en las que el acumulador resultaba alterado. Además el valor del bit de acarreo es respetado, aunque no pasa lo mismo con las demás banderas.

SETeando y RESeteando un bit

Existen otras dos instruciones que permiten modificar el valor lógico de un determinado bit en cualquier registro o posición de memoria que direccionemos con HL, IX o IY.

SET lo que hace es poner a uno el bit que indiquemos fuera cual fuera el valor anterior y respetando los bits contiguos. La sintaxis es similar a la de BIT; por ejemplo, para poner a uno el bit 3 del contenido de la dirección de memoria a la que apunta HL habría que hacer SET 3,(HL). Esto equivaldría a hacer LD A,32 más OR (HL) y más LD (HL),A; como podemos ver estas instrucciones ahorran bastante, además de reservar el valor que tuviera el acumulador.

RES pone a cero el bit que especifiquemos de forma similar a BIT y SET. Para poner a cero el bit 0 del acumulador haríamos RES 0,A, lo que es equivalente a hacer AND 254.

Las instrucciones de manipulación de bits, aunque no son tan utilizadas como las de carga o las aritméticas, cumplen un papel importante a la hora de hacer más cómodos y rápidos ciertos cometidos concretos. Hay que tener siempre presente que existen, de forma que puedan sernos útiles cuando las necesitemos.

Tiempos de ejecución

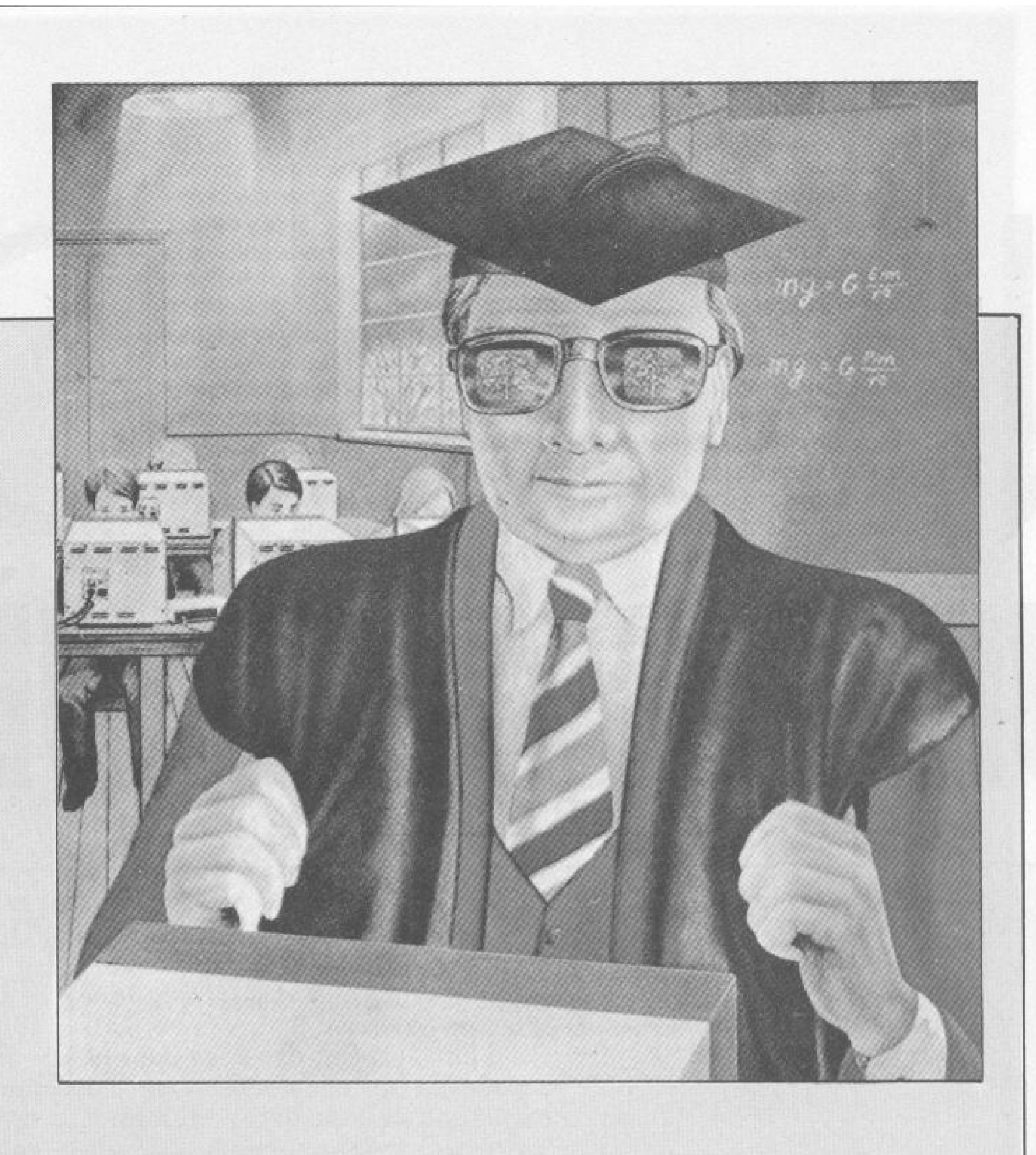
El tiempo de ejecución puede resultar una cualidad importante en un programa y a menudod es imprescindible su cálculo para ciertas rutinas que deben compenetrarse entre sí a la perfección. Por ejemplo, no podrían funcionar las rutinas de

```
NOTACION

r y r' = A,B,C,D,E,H o L
d = desplazamiento (-128 a 127)
n = valor entre 0 y 255 (1 byte)
nn = valor entre 0 y 65535 (2 bytes)
rr = BC,DE,HL o SP
qq = AF,BC,DE o HL
rX = BC,DE,IX o SP
rY = BC,DE,IY o SP
b = bit (0-7)
cc = condición
n' = 0,8,16,24,32,40,48 o 56

BANDERAS

* = no alterada
X= alterada según el resultado de la operación
0 = puesto a cero
1 = puesto a uno
i = toma el valor de IFF (habilitación de interrupción)
? = desconocido
2 = cero si BC-1=0, uno en caso contrario
3 = uno si A=(HL), cero en caso contrario
4 = uno si B-1=0, cero en caso contrario
```



Mnemònico	C	Z	P/V	S	N	H	Nùm. bytes	Nùm. est.T
ADD HL,rr	X	*	*	*	0	?	1	11
ADC HL,rr	X	X	X	X	0	?	2	15
SBC HL,rr	X	X	X	X	1	?	2	15
ADD IX,rX	X	*	*	*	0	?	2	15
ADD IY,rY	X	*	*	*	0	?	2	15
INC rr	X	X	X	X	X	X	1	6
INC IX	X	X	X	X	X	X	2	10
INC IY	X	X	X	X	X	X	2	10
DEC rr	X	X	X	X	X	X	1	6
DEC IX	X	X	X	X	X	X	2	10
DEC IY	X	X	X	X	X	X	2	10
RLCA	X	*	*	*	0	0	1	4
RLA	X	*	*	*	0	0	1	4
RRCA	X	*	*	*	0	0	1	4
RRA	X	*	*	*	0	0	1	4
RLC r	X	X	X	X	0	0	2	8
RLC (HL)	X	X	X	X	0	0	2	15
RLC (IX+d)	X	X	X	X	0	0	4	23
RLC (IY+d)	X	X	X	X	0	0	4	23
RL s	X	X	X	X	0	0		
RRC s	X	X	X	X	0	0		
RR s	X	X	X	X	0	0		
SLA s	X	X	X	X	0	0		
SRA s	X	X	X	X	0	0		
SRL s	X	X	X	X	0	0		
RLD	*	X	X	X	0	0	2	18
RRD	*	X	X	X	0	0	2	18

s es cualquiera de r, (HL), (IX+d) o (IY+d) como en RLC.

grabación y carga cassette si no coincidieran en ambas los tiempos de los bucles de retardo.

El cálculo de los tiempos de ejecución de una instrucción resulta siempre que sepamos el número de pasos de reloj o estados temporales que necesita dicha instrucción y la velocidad a la que corre el reloj en nuestro ordenador. En el Spectrum la velocidad de reloj es de 3.5 MHz, es decir, que en un segundo da tres millones y medio de impulsos, y el número de estados temporales de cada instrucción, así como la forma en que las banderas se ven afectadas, podéis consultarlo en la tabla que acompaña a estas líneas.

Si en un segundo nuestro reloj da 3.500.000 impulsos, quiere decir que el tiempo de cada ciclo es de $1/3.5*10^6$, osea, aproximadamente .000000286 segundos ó 286 nanosegundos.

LD A,I	* X i X 0 0	2	9
LD A,R	* X i X 0 0	2	9
LD I,A	* * * * * *	2	9
LD R,A	* * * * * *	2	9
LD rr,nn	* * * * * *	3	10
LD IX,nn	* * * * * *	4	14
LD IY,nn	* * * * * *	4	14
LD HL,(nn)	* * * * * *	3	16
LD dd,(nn)	* * * * * *	4	20
LD IX,(nn)	* * * * * *	4	20
LD IY,(nn)	* * * * * *	4	20
LD (nn),HL	* * * * * *	3	16
LD (nn),dd	* * * * * *	4	20
LD (nn),IX	* * * * * *	4	20
LD (nn),IY	* * * * * *	4	20
LD SP,HL	* * * * * *	1	6
LD SP,IX	* * * * * *	2	10
LD SP,IY	* * * * * *	2	10
PUSH qq	* * * * * *	1	11
PUSH IX	* * * * * *	2	15
PUSH IY	* * * * * *	2	15
POP qq	* * * * * *	1	10
POP IX	* * * * * *	2	14
POP IY	* * * * * *	2	14

Mnemònico	Banderas C Z P/V S N H	Nùm. bytes	Nùm. est.T	
JP nn	* * * * * *	3	10	
JP cc,nn	* * * * * *	3	10	
JR d	* * * * * *	2	12	Si cc es falsa
			7	Si cc es verdadera
JR C,d	* * * * * *	2	12	Si cc es falsa
			7	Si cc es verdadera
JR NC,d	* * * * * *	2	12	Si cc es falsa
			7	Si cc es verdadera
JR Z,d	* * * * * *	2	12	Si cc es falsa
			7	Si cc es verdadera
JR NZ,d	* * * * * *	2	12	Si cc es falsa
			7	Si cc es verdadera
JP (HL)	* * * * * *	1	4	
JP (IX)	* * * * * *	2	8	
JP (IY)	* * * * * *	2	8	
DJNZ d	* * * * * *	2	8	Si B=0
			13	Si B<>0
CALL nn	* * * * * *	3	17	
CALL cc,nn	* * * * * *	3	10	Si cc es falsa
			17	Si cc es verdadera
RET	* * * * * *	1	10	
RET cc	* * * * * *	1	5	Si cc es falsa
			11	Si cc es verdadera
RETI	* * * * * *	2	14	
RETN	* * * * * *	2	14	
RST n'	* * * * * *	1	11	
IN A,(n)	* * * * * *	2	10	
IN r,(C)	* X X X 0 X	2	11	
INI	* 4 ? ? 1 ?	2	15	
INIR	* 1 ? ? 1 ?	2	20	Si B<>0
			15	Si B=0
IND	* 4 ? ? 1 ?	2	15	
INDR	* 1 ? ? 1 ?	2	20	Si B<>0
			15	Si B=0
OUT (n),A	* * * * * *	2	11	
OUT (C),r	* * * * * *	2	12	
OUTI	* 4 ? ? 1 ?	2	15	
OTIR	* 1 ? ? 1 ?	2	20	Si B<>0
			15	Si B=0
OUTD	* 4 ? ? 1 ?	2	15	
OTDR	* 1 ? ? 1 ?	2	20	Si B<>0
			15	Si B=0

Por ejemplo, la siguiente rutina:

Instrucciones	n.º estados T	n.º veces ejec.	Total
XOR A	7	1	7
LD B,10	7	1	7
BUCLE ADD A,B	4	10	40
DJNZ BUCLE	8 si B=0	1	8
	12 si B<>0	9	108
RET	10	1	10
		Total	180

nos da un total de 180 estados temporales, lo que equivale a 180/3.5*10°, o aproximadamente .0000514 segundos, es decir, bastante poco tiempo.

Con el estudio de las instrucciones de manipulación de bits damos en este capítulo casi por acabada lo que ha pretendido ser una visión general del juego de instrucciones del Z-80. En capítulos sucesivos intentaremos terminar de adentrar al lector en la estructura lógica del Spectrum aclarando los temas más esenciales para hacerle llegar al punto en el que pueda sacarle todo el partido a su ordenador.

BIT b,r	* X ? ? 0 1	2	8
BIT b,(HL)	* X ? ? 0 1	2	12
BIT b,(IX+d)	* X ? ? 0 1	4	20
BIT b,(IY+d)	* X ? ? 0 1	4	20
SET b,r	* * * * * *	2	8
SET b,(HL)	* * * * * *	2	15
SET b,(IX+d)	* * * * * *	4	23
SET b,(IY+d)	* * * * * *	4	23
RES b,s	s es cualquiera de r, (HL), (IX+d) o (IY+d) como en SET.		

Mnemònico	Banderas C Z P/V S N H	Nùm. bytes	Nùm. est.T
LD r,r'	* * * * * *	1	4
LD r,n	* * * * * *	2	7
LD r,(HL)	* * * * * *	1	7
LD r,(IX+d)	* * * * * *	3	19
LD r,(IY+d)	* * * * * *	3	19
LD (HL),r	* * * * * *	1	7
LD (IX+d),r	* * * * * *	3	19
LD (IY+d),r	* * * * * *	3	19
LD (HL),n	* * * * * *	2	10
LD (IX+d),n	* * * * * *	4	19
LD (IY+d),n	* * * * * *	4	19
LD A,(BC)	* * * * * *	1	7
LD A,(DE)	* * * * * *	1	7
LD A,(nn)	* * * * * *	3	13
LD (BC),A	* * * * * *	1	7
LD (DE),A	* * * * * *	1	7
LD (nn),A	* * * * * *	3	13

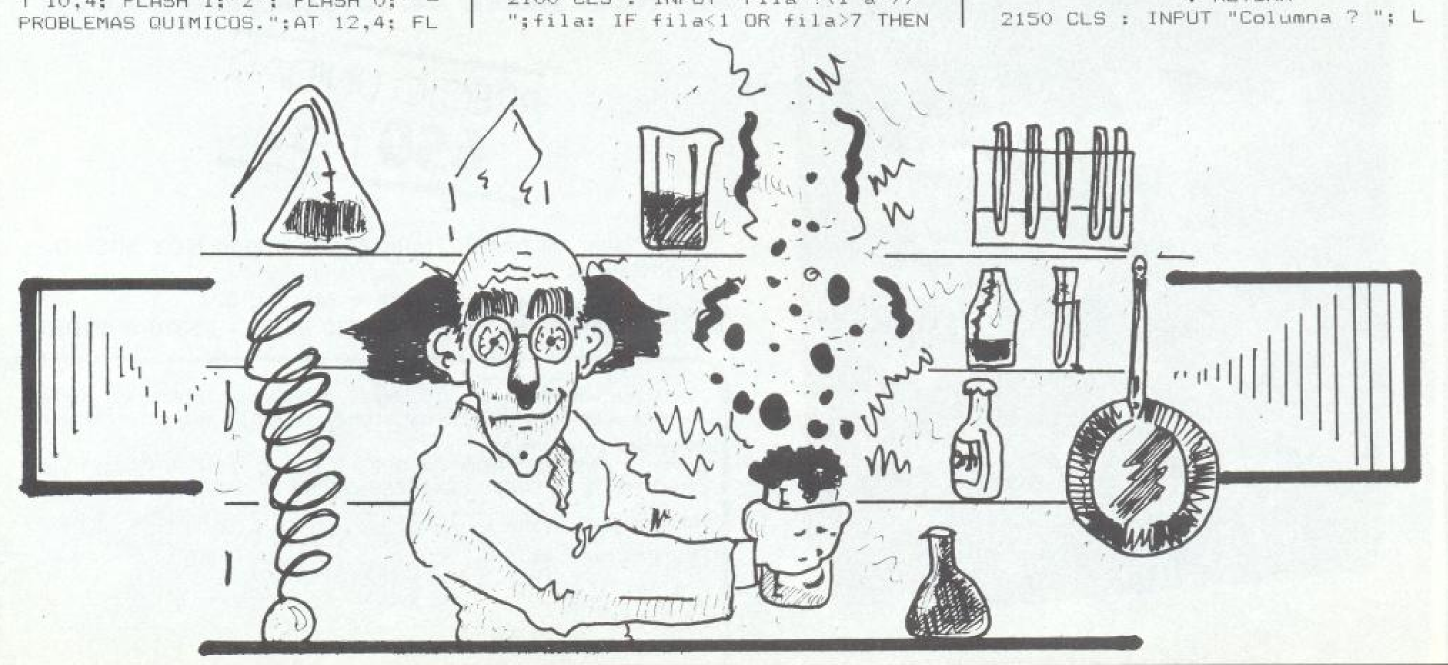
SISTEMA PERIODICO

El programa consta de tres partes:

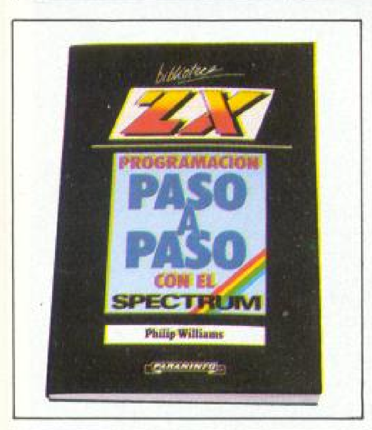
- **Sistema Periódico. Esta opción nos da a elegir entre filas, columnas o grupos. Las filas están numeradas de 1 a 7 y corresponden a cada uno de los siete períodos. Las columnas se nombran 1a, 2a,... o 1b, 2b,... o 0 para el grupo de los gases nobles. En la opción de grupos aparecen casi todos los grupos convencionales como alcalinos, alcalino-térreos, etc.**
- **Problemas químicos. Esta opción comprende dos casos: estructuras electrónicas y pesos moleculares. La estructura se halla por la regla de Moeller sin ninguna complicación ni error.**
- **Búsqueda de elementos. Se realiza por número atómico, nombre o símbolo químico. En el primer caso es mucho más rápido, ya que no tiene que leer los nombres, sino que va directamente a la línea DATA donde se encuentran. Si buscamos los datos por el nombre tendremos que poner la primera letra con mayúscula y las demás con minúsculas. La búsqueda por símbolo requiere que éste se escriba en la forma tradicional, la primera letra con mayúscula y la segunda con minúscula.**

Fernando Arderius Martín

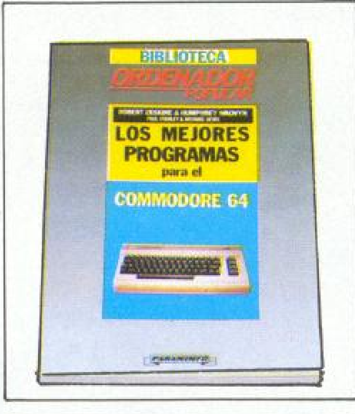
```
  10 REM FERNANDO ARDERIUS MARIN
  20 BORDER 1: PAPER 3: CLS
  25 INK 0
  30 CLEAR 64999: DIM c$(3.16)
  40 FOR x=65000 TO 65079
  50 READ a: POKE x,a: NEXT x
  60 DATA 237,75,176,92,121.254,
32,56
  70 DATA 1,201,120,254,21,56.1,
201
  80 DATA 205,158,14.197.6,0,9,1
93
  85 DATA 229,229,120,60.205,158
,14,6
  90 DATA 0,9,209,1,0,7,9,235
 100 DATA 9,229,6,4,126,18,21,18
 105 DATA 21,37,16.248,209,6,4.1
26
 110 DATA 18.21,18.21,37,16,248.
225
 120 DATA 124,15.15,15,230.3,246
,88
 130 DATA 103,126,1,32,0,9,119.2
01
 200 CLS : PRINT  INVERSE 1;"
   SISTEMA PERIODICO.          ":
INVERSE 0
 210 PRINT AT 8,4: FLASH 1;"1";
FLASH 0;" -SISTEMA PERIODICO.";A
T 10,4: FLASH 1;"2": FLASH 0;" -
PROBLEMAS QUIMICOS.";AT 12,4; FL
ASH 1;"3": FLASH 0;" -BUSQUEDA D
E ELEMENTOS."
 220 PRINT AT 21,0; PAPER 0; INK
 3;"   FERNANDO ARDERIUS MARIN
.   "
 230 PAUSE 0: LET i$=INKEY$
 240 IF CODE i$<49 OR CODE i$>51
 THEN  GO TO 230
 250 BEEP .1,30: GO TO 2000*VAL
i$
2000 REM sistema periodico.
2010 CLS : PRINT  INVERSE 1;"
    SISTEMA PERIODICO.          ":
INVERSE 0
2020 PRINT AT 8,4; FLASH 1;"1";
FLASH 0;"- Elementos por filas."
;AT 10,4; FLASH 1;"2"; FLASH 0;"
- Elementos por columnas.";AT 12
,4; FLASH 1;"3"; FLASH 0;"- Elem
entos por grupos."
2030 PRINT AT 19,10;"s para sali
r.";AT 21,0; PAPER 0;"
                              "
2040 PAUSE 0: LET i$=INKEY$: IF
i$="s" THEN  GO TO 200
2050 IF CODE i$<49 OR CODE i$>51
 THEN  GO TO 2040
2060 BEEP .1,30: GO TO 2050+50*V
AL i$
2100 CLS : INPUT "Fila ?(1 a 7)
";fila: IF fila<1 OR fila>7 THEN
  GO TO 2100
2105 PRINT AT 9,10;"La fila ";fi
la;" es:": GO SUB 2120+fila: PRI
NT AT 11,0; PAPER 0; INK 7;w$
2110 BEEP .1,20: INPUT "Otra ?(s
/n) "; LINE x$: IF x$="s" OR x$=
"S" THEN  GO TO 2100
2115 IF x$="n" OR x$="N" THEN  G
O TO 2000
2120 GO TO 2110
2121 LET w$="H,He
               ": RETURN
2122 LET w$="Li,Be,B,C,N,O,F,Ne
               ": RETURN
2123 LET w$="Na,Mg,Al,Si,P,S,P,S
,Cl,Ar         ": RETURN
2124 LET w$="K,Ca,Sc,Ti,V,Cr,Mn,
Fe,Co,Ni,Cu, Zn,Ga,Ge,As,Se,Br,K
r              ": RETURN
2125 LET w$="Rb,Sr,Y,Zr,Nb,Mo,Tc
,Ru,Rh,Pd,Ag,Cd,In,Sn,Sb,Te,I,Xe
               ": RETURN
2126 LET w$="Cs,Ba,La,Ce,Pr,Nd,P
m,Sm,Eu,Gd,Tb,Dy,Ho,Er,Tm,Yb,Lu,
Hf,Ta,W,Re,Os,Ir,Pt,Au,Hg,Tl,Pb,
Bi,Po,At,Rn  ": RETURN
2127 LET w$="Fr,Ra,Ac,Th,Pa,U,Np
,Pu,Am,Cm,Bk,Cf,Es,Fm,Md,No,Lw
               ": RETURN
2150 CLS : INPUT "Columna ? "; L
```







P.V.P. 750 PTAS.
(IVA INCLUIDO)
Descubre los misterios de la programación de una forma sencilla, con ejemplos, programas y organigramas.
(110 páginas, tamaño 13,5 x 21)



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Con utilidades, juegos explosivos y gráficos dinámicos que lleva al BASIC hasta el mejor aprovechamiento de sus posibilidades.
(200 páginas, tamaño 15,5 x 21,5).



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Un libro especialmente dedicado a los que se inician por vez primera en el mundo del Spectrum.
(100 páginas, tamaño 13,5 x 21).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Una inestimable ayuda que complementará la que proporciona el manual del ordenador.
(108 páginas tamaño 13,5 x 21,5).



P.V.P. 900 PTAS.
(IVA INCLUIDO)
Un compendio de los programas más diversos con los que podrá aprender jugando las importantnes características del BASIC.
(258 páginas, tamaño 15,5 x 21,5).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Muestra una visión más completa del correcto funcionamiento del juego de instrucciones del C-64.
(108 páginas, tamaño 13,5 x 21,5).

CUPON DE PEDIDO

enviar a:
infodis, s.a.
C/BRAVO MURILLO, 377
28020 MADRID

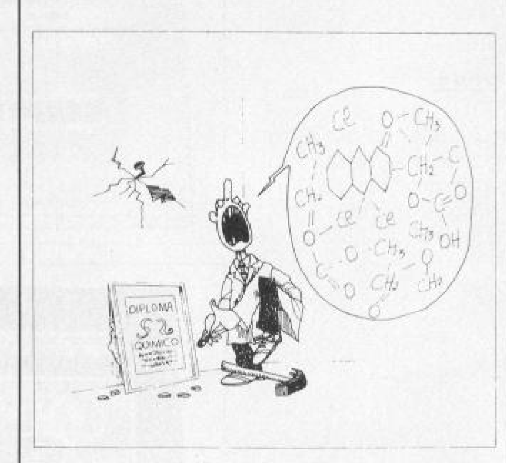
COPIE O RECORTE ESTE BOLETIN DE PEDIDO.

DESEO RECIBIR LOS SIGUIENTES TITULOS:
15 HORAS CON EL SPECTRUM (P.V.P. 750) ☐
LOS MEJORES PROGRAMAS PARA EL ZX SPECTRUM (P.V.P. 900) ☐
LOS MEJORES PROGRAMAS PARA EL COMMODORE 64 (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL I (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL II (P.V.P. 800) ☐
(más 100 ptas. de gastos de envío).

El importe lo abonaré POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ American Express ☐ Visa ☐ Interbank ☐
Número de mi tarjeta: ____________________

NOMBRE ____________________
CALLE ____________________
CIUDAD ____________________
PROVINCIA ____________________ C. P. __________

```
INE p$: IF LEN p$>2 THEN  GO TO
2150
2160 IF p$="1a" THEN  LET w$="H
LiNaK RbCsFr": GO TO 2182
2161 IF p$="2a" THEN  LET w$="
BeMgCaSrBaRa": GO TO 2182
2162 IF p$="3a" THEN  LET w$="
   ScY LaAc": GO TO 2182
2163 IF p$="4a" THEN  LET w$="
  TiZrHfCeTh": GO TO 2182
2164 IF p$="5a" THEN  LET w$="
    V NbTaPrPa": GO TO 2182
2165 IF p$="6a" THEN  LET w$="
  CrMoW NdU ": GO TO 2182
2166 IF p$="7a" THEN  LET w$="
  MnTcRePmNp": GO TO 2182
2167 IF p$="8a" THEN  LET c$(1)=
"        FeRuOsSmPu": LET c$(2)="
     CoRhIrEuAm": LET c$(3)="
   NiPdPtGdCm": PRINT AT 7,7;"La
 columna 8a es:": FOR x=1 TO 3:
LET w$=c$(x): FOR y=1 TO LEN w$
STEP 2: PRINT AT 10+(y/2),11+(x*
2); PAPER 0; INK (4+x);w$(y TO y
+1): NEXT y: NEXT x: BEEP .1,20:
 GO TO 2190
2168 IF p$="1b" THEN  LET w$="
     CuAgAuTbBk": GO TO 2182
2169 IF p$="2b" THEN  LET w$="
     ZnCdHgDyCf": GO TO 2182
2170 IF p$="3b" THEN  LET w$="
B AlGaLnTlHoTs": GO TO 2182
2171 IF p$="4b" THEN  LET w$="
C SiGeSnPbErFm": GO TO 2182
2172 IF p$="5b" THEN  LET w$="
N P AsSbBiTmMd": GO TO 2182
2173 IF p$="6b" THEN  LET w$="
O S SeTePoYbNo": GO TO 2182
2174 IF p$="7b" THEN  LET w$="
F ClBrI AtLuLw": GO TO 2182
2175 IF p$="0" THEN  LET w$="HeN
eArKrXeRnHa??": GO TO 2182
2180 PRINT "    Los grupos posib
les son:"''; PAPER 2; INK 6;"1a2
a3a4a5a6a8a1b2b3b4b5b6b7b y 0":
BEEP .1,-30: PRINT )0;"
 pulsa tecla.          ": PAUSE
0: GO TO 2150
2182 CLS : PRINT AT 7,7;"La colu
mna ";p$;" es:"
2184 FOR x=1 TO LEN w$ STEP 2
2186 PRINT AT 10+(x/2),15; PAPER
 0; INK 7;w$(x TO x+1)
2188 NEXT x: BEEP .1,20
2190 INPUT "Otra columna ?(s/n)
"; LINE x$: IF x$="s" OR x$="S"
THEN  GO TO 2150
2192 IF x$="n" OR x$="N" THEN  G
O TO 2000
2194 GO TO 2190
2200 CLS : INPUT "Grupo ? "; LIN
E g$
2205 IF g$="alcalinos" OR g$="Al
calinos" THEN  LET w$="H,Li,Na,K
,Rb,Cs,Fr": GO TO 2300
2206 IF g$="alcalino-terreos" OR
 g$="Alcalino-terreos" THEN  LET
 w$="Be,Mg,Ca,Sr,Ba,Ra": GO TO 2
300
2207 IF g$="gases nobles" OR g$=
"Gases nobles" THEN  LET w$="He,
Ne,Ar,Kr,Xe,Rn": GO TO 2300
2208 IF g$="halogenos" OR g$="Ha
logenos" THEN  LET w$="F,Cl,I,At
": GO TO 2300
2209 IF g$="tierras raras" OR g$
="Tierras raras" THEN  GO SUB 30
00: LET w$=k$: GO TO 2300
2210 IF g$="lantanidos" OR g$="L
antanidos" THEN  GO SUB 3000: LE
T w$=k$(1 TO LEN k$/2): GO TO 23
00
2211 IF g$="actinidos" OR g$="Ac
tinidos" THEN  GO SUB 3000: LET
w$=k$(LEN k$/2 TO LEN k$): GO TO
 2300
2212 IF g$="terreos" OR g$="Terr
eos" THEN  LET w$="B,Al,Ga,In,Tl
": GO TO 2300
2213 IF g$="Carbonoideos" OR g$=
"carbonoideos" THEN  LET w$="C,S
i,Ge,Sn,Pb": GO TO 2300
2214 IF g$="nitrogenoideos" OR g
$="Nitrogenoideos" THEN  LET w$=
"N,P,As,Sb,Bi": GO TO 2300
2215 IF g$="anfigenos" OR g$="An
figenos" THEN  LET w$="O,S,Se,Te
,Po": GO TO 2300
2220 FOR x=2 TO 12: PRINT AT x,0
; PAPER 2;"
                 ": NEXT x
2221 PRINT AT 0,0;"Los grupos po
sibles son:"; PAPER 2; INK 6;''"
Alcalinos"'"Alcalino-terreos"'"T
erreos"'"Carbonoideos"'"Nitrogen
oideos"'"Anfigenos"'"Halogenos"'
"Gases nobles"'"Tierras raras"'"
Lantanidos"'"Actinidos"
```




```
2230 BEEP .1,-30: PRINT )0;TAB 1
0;"pulsa tecla.": PAUSE 0: CLS :
 GO TO 2200
2300 CLS : LET fil=0: LET col=10
: LET e=1
2310 FOR x=1 TO LEN w$
2315 LET fil=fil+1: LET e=-e
2320 BEEP .01,e*(INT (RND*30)):
PRINT  PAPER 0; INK 7;AT col,fil
;w$(x): IF fil>30 THEN  LET fil=
0: LET col=col+1
2330 NEXT x
2340 PRINT AT 6,0;"El grupo de l
os ";g$;" es:"
2350 INPUT "Otro ?(s/n) "; LINE
x$: IF x$="s" OR x$="S" THEN  GO
 TO 2200
2360 IF x$="n" OR x$="N" THEN  G
O TO 2000
2370 GO TO 2350
3000 LET k$=" La,Ce,Pr,Nd,Pm,Sm,
Eu,Gd,Tb,Dy, Er,Tm,Yb,Lu,Ac,Th,P
a,U ,Np,Pu,Am,Cm,Bk,Cf,Es,Fm,Md,
No,Lw": RETURN
4000 REM problemas quimicos
4010 CLS : PRINT AT 0,0; INVERSE
 1;"      PROBLEMAS QUIMICOS.
    "; INVERSE 0
4020 PRINT AT 9,4; FLASH 1;"1";
FLASH 0;"- ESTRUCTURA ELECTRONIC
A.";AT 11,4; FLASH 1;"2"; FLASH
0;"- PESOS MOLECULARES.";AT 19,1
0;"s para salir.";AT 21,0; PAPER
 0;"
     "
4030 PAUSE 0: LET i$=INKEY$: IF
i$="s" OR i$="S" THEN  GO TO 200
4040 IF CODE i$<49 OR CODE i$>50
 THEN  GO TO 4030
4050 BEEP .1,30: GO TO 4050+200*
VAL i$
4250 CLS : INPUT "Numero atomico
? ";n
4255 IF n<0 OR n>105 THEN  GO TO
 4250
4260 DIM x$(19,2): DIM d(19): LE
T cont=0: LET fil=0: LET color=0
: LET col=0
4270 LET e$="1223343454564567567
": LET z$="sspspsdpsdpsfdpsfdp":
 LET d$="1131315315317531753"
4275 FOR x=1 TO 19
4280 LET x$(x)=e$(x)+z$(x)
4285 LET d(x)=VAL d$(x)*2
4287 LET cont=cont+d(x)
4290 IF cont=n THEN  GO TO 4300
4295 IF cont>n THEN  LET sobra=d
(x): GO TO 4299
4298 NEXT x
4299 LET d(x)=n-(cont-sobra)
4300 LET e=1: FOR y=1 TO x
4302 LET fil=fil+4: LET e=-e
4305 BEEP .01,e*(INT (RND*30)):
PRINT AT 7+col,fil; INK color;x$
(y); INK color+5;d(y): IF fil>25
 THEN  LET fil=0: LET col=col+1
4310 NEXT y
4315 PRINT AT 2,1;"-La estructur
a electronica del"'"elemento de
num. atomico ";n;" es:"
4320 INPUT "Otra ?(s/n) ": LINE
q$: IF q$="s" OR q$="S" THEN  GO
 TO 4250
4330 IF q$="n" OR q$="N" THEN  C
LEAR : GO TO 4000
4340 GO TO 4320
4450 CLS : INPUT "Compuesto ? ";
 LINE y$: LET l=LEN y$: IF (l/3)
<>INT (l/3) THEN  GO TO 4450
4455 DIM f(5): DIM k(5): IF l>15
 THEN  GO TO 4450
4460 LET o=0
4461 FOR x=1 TO l STEP 3: LET o=
o+1
4462 IF y$(x+1)=" " THEN  LET t$
=y$(x): GO TO 4464
4463 LET t$=y$(x)+y$(x+1)
4464 FOR g=1 TO 105
4465 RESTORE 9000+g
4466 READ n,n$,m$,m,v$,r,c2,v,f2
,e
4467 IF n$=t$ THEN  GO TO 4470
4469 NEXT g: GO TO 4450
4470 LET f(o)=INT (m+.5)
4472 LET k(o)=VAL y$(x+2)
4475 NEXT x
4477 LET peso=0
4480 FOR x=1 TO o+1
4485 LET peso=peso+f(x)*k(x)
4490 NEXT x
4495 BEEP .1,20: PRINT AT 3,0;"-
La masa del compuesto ";y$;'"es:
"'"       ";peso;" u.m.a."
4500 INPUT "Otra ?(s/n) "; LINE
x$: IF x$="s" OR x$="S" THEN  GO
 TO 4450
4510 IF x$="n" OR x$="N" THEN  G
O TO 4000
6000 REM busqueda de elementos.
6010 CLS
6020 PRINT  INVERSE 1;"       BUSQ
UEDA DE ELEMENTOS.        "; INVERS
E 0
6030 PRINT AT 8,2; FLASH 1;"1";
FLASH 0;" -BUSQUEDA POR NUM.ATOM
ICO.";AT 10,2; FLASH 1;"2"; FLAS
H 0;" -BUSQUEDA POR NOMBRE.";AT
12,2; FLASH 1;"3"; FLASH 0;" -BU
SQUEDA POR SIMB.QUIMICO."
6040 PRINT AT 19,10;"s para sali
r.";AT 21,0;"[redacted]"
6050 PAUSE 0: LET i$=INKEY$: IF
i$="s" OR i$="S" THEN  GO TO 200
6060 IF CODE i$<49 OR CODE i$>51
 THEN  GO TO 6050
6070 CLS : BEEP .1,30: GO TO 700
0+100*VAL i$
7100 INPUT "N.atomico? ";n
7110 IF n<0 OR n>105 THEN  GO TO
 7100
7120 RESTORE 9000+n
7130 READ n,n$,m$,m,v$,r,c2,v,f2
,e
7140 GO TO 8000
7200 INPUT "Nombre? "; LINE g$
7210 FOR x=1 TO 105
7215 RESTORE 9000+x
```

```
7220 READ n,n$,m$,m,v$,r,c2,v,f2
,e: IF g$=m$ THEN  GO TO 8000
7230 NEXT x
7240 PRINT )0;"     No existe ese
 elemento                pulsa t
ecla": PAUSE 0: CLS : GO TO 7200

7400 INPUT "Simbolo? "; LINE s$
7410 FOR x=1 TO 105
7415 RESTORE 9000+x
7420 READ n,n$,m$,m,v$,r,c2,v,f2
,e: IF s$=n$ THEN  GO TO 8000
7430 NEXT x
7440 PRINT )0;"      No existe es
e simbolo                pulsa t
ecla": PAUSE 0: CLS : GO TO 7400
8000 BORDER 0: PAPER 0: CLS : IN
K 7: FOR x=0 TO 3: PRINT AT x,0;
 PAPER 1;"
           ": NEXT x
8010 FOR x=4 TO 8: PRINT AT x,0;
 PAPER 4;"              ";AT x,22;
"            ";AT x,11; PAPER 2;"
           ": NEXT x
8020 LET c=7-LEN m$/2
8030 POKE 23729,1
8040 FOR f=1 TO LEN m$
8050 POKE 23728,(f+c)*2
8060 PRINT  PAPER 1;AT 1,(f+c)*2
;m$(f)
8070 RANDOMIZE USR 65000
8075 NEXT f
8080 POKE 23729,6
8090 LET c=8-LEN n$
8100 FOR f=1 TO LEN n$

8110 POKE 23728,(f+c)*2
8120 INK 7: PRINT  PAPER 2;AT 6,
(f+c)*2;n$(f)
8130 RANDOMIZE USR 65000
8140 NEXT f
8150 PRINT  INK 0; PAPER 4;AT 5,
0;"NUMERO";AT 6,0;"ATOMICO:";n;A
T 5,22;"PESO ATOMI";AT 6,22;"CO:
";m
8160 FOR x=9 TO 12: PRINT AT x,0
; PAPER 0;"
            ": NEXT x
8170 PRINT AT 9,12; PAPER 0; INK
 7;"VALENCIAS";AT 11,16-LEN v$/2
;v$
8180 FOR x=13 TO 16: PRINT AT x,
0; PAPER 7;"               ";AT
 x,16; PAPER 1;"
": NEXT x
8190 FOR x=17 TO 20: PRINT AT x,
0; PAPER 1;"               ";AT
 x,16; PAPER 7;"
": NEXT x
8200 PRINT AT 13,0; PAPER 7; INK
 1;"RADIO ATOMICO";AT 14,5;r;AT
17,16;"AFINIDAD";AT 18,21;f2
8210 PRINT AT 13,16; PAPER 1; IN
K 7;"VOLUMEN ATOMICO";AT 14,21;v
;AT 17,0;"RADIO COVALENTE";AT 18
,5;c2

8220 PRINT AT 21,0; PAPER 7; BRI
GHT 1;"
      "
8230 PRINT AT 21,0; INK 0; PAPER
 7; BRIGHT 1;"ELECTRONEGATIVIDAD
:";e
```

```
        S e l e n i o

NUMERO                    PESO ATOMI
ATOMICO:34       S e      CO:78.96

            VALENCIAS

             6/4/-2

RADIO ATOMICO    VOLUMEN ATOMICO
     1.4               16.5

RADIO COVALENTE  AFINIDAD
     1.16              -203

ELECTRONEGATIVIDAD:2.4
```

```
8240 INPUT "Otro ?(s/n) "; LINE
x$: IF x$="s" OR x$="S" THEN  PA
PER 3: CLS : BORDER 1: INK 0: GO
 TO 6000
8250 IF x$="n" OR x$="N" THEN  P
APER 3: CLS : BORDER 1: INK 0: G
O TO 200
8260 GO TO 8240

9001 DATA 1,"H","Hidrogeno",1.00
8,"+1,-1",0,0.37,14.1,-74.5,2.1
9002 DATA 2,"He","Helio",4,"0",0
,0.83,31.8,21.2,0
9003 DATA 3,"Li","Litio",6.94,"1
",1.55,1.34,13.4,-59.8,1
9004 DATA 4,"Be","Berilio",9.01,
"2",1.12,0.90,5.0,-37.6,1.5
9005 DATA 5,"B","Boro",10.81,"3"
,0.98,0.82,4.6,-17.3,2
9006 DATA 6,"C","Carbono",12.01,
"+-4/2",0.914,0.77,5.3,-122.3,2.
5
9007 DATA 7,"N","Nitrogeno",14.0
1,"5/+-3/2/4",0.92,0.75,17.3,20.
1,3
9008 DATA 8,"O","Oxigeno",16,"-1
/-2",0,0.73,14.0,-141.3,3.5
9009 DATA 9,"F","Fluor",19,"-1",
0,0.72,17.1,-337.4
9010 DATA 10,"Ne","Neon",20.18,"
0",0,1.31,16.8,28.9,0
9011 DATA 11,"Na","Sodio",22.99,
"1",1.90,1.54,23.7,-52.2,0.9
9012 DATA 12,"Mg","Magnesio",24.
31,"2",1.60,1.30,14.0,21.2,1.2
9013 DATA 13,"Al","Aluminio",26.
98,"3",1.43,1.18,10.0,-19.3,1.5
9014 DATA 14,"Si","Silicio",28.0
9,"+-4",1.32,1.11,12.1,-131,1.8
9015 DATA 15,"P","Fosforo",30.97
,"5/+-3",1.28,1.06,17.0,-68.5,2.
1

9016 DATA 16,"S","Azufre",32.06,
"6/-2/4",1.27,1.02,15.5,-196.8,2
.5

9017 DATA 17,"Cl","Cloro",35.45,
"+-1/7/5",0,0.99,18.7,-349.2,3
9018 DATA 18,"Ar","Argon",39.95,
"0",0,1.74,24.2,35.7,0
9019 DATA 19,"K","Potasio",39.10
,"1",2.35,1.96,45.3,-45.4,0.8
9020 DATA 20,"Ca","Calcio",40.08
,"2",1.97,1.74,29.9,1.86,1
9021 DATA 21,"Sc","Escandio",44.
96,"3",1.62,1.44,15.0,70.5,1.3
9022 DATA 22,"Ti","Titanio",47.9
0,"4/3",1.47,1.38,10.6,1.93,1.5
9023 DATA 23,"V","Vanadio",50.94
,"5/4/3/2",1.34,0,8.35,-60.8,1.6
9024 DATA 24,"Cr","Cromo",52,"3/
6/2",1.30,0,7.23,93.5,1.6
9025 DATA 25,"Mn","Manganeso",54
.94,"2/7/3/4/6",1.35,0,7.39,93.5
,1.5
9026 DATA 26,"Fe","Hierro",55.85
,"3/2",1.26,0,7.1,44.5,1.8
9027 DATA 27,"Co","Cobalto",58.9
3,"2/3",1.25,0,6.7,-102,1.8
9028 DATA 28,"Ni","Niquel",58.71
,"2/4/3",1.24,0,6.6,-156,1.8
9029 DATA 29,"Cu","Cobre",63.54,
"2/1",1.28,1.38,7.1,-173,1.9
9030 DATA 30,"Zn","Cinc",65.37,"
2",1.38,1.31,9.2,8.7,1.6
9031 DATA 31,"Ga","Galio",69.72,
"3",1.41,1.26,11.8,-35.3,1.6
9032 DATA 32,"Ge","Germanio",72.
59,"+-4/2",1.37,1.22,13.6,-139,1
.8
9033 DATA 33,"As","Arsenico",74.
92,"5/+-3",1.39,1.19,13.1,-103,2
9034 DATA 34,"Se","Selenio",78.9
6,"6/4/-2",1.40,1.16,16.5,-203,2
.4

9035 DATA 35,"Br","Bromo",79.91,
"+-1/5",0,1.14,23.5,-324.1,2.8
9036 DATA 36,"Kr","Cripton",83.8
0,"0",0,1.89,32.2,40.5,0
9037 DATA 37,"Rb","Rubidio",85.4
7,"1",2.48,2.11,55.9,-37.6,0.8
9038 DATA 38,"Sr","Estroncio",87
.62,"2",2.15,1.92,33.7,145,1
9039 DATA 39,"Y","Ytrio",88.91,"
3",1.78,1.62,19.8,38.6,1.3
9040 DATA 40,"Zr","Circonio",91.
22,"4/3/2",1.60,1.48,14.1,-43.5,
1.4
9041 DATA 41,"Nb","Niobio",92.91
,"5/3/4/2",1.46,0,10.8,-109,1.6
9042 DATA 42,"Mo","Molibdeno",95
.94,"6/3/5/4",1.39,0,9.4,-114,1.
8
9043 DATA 43,"Tc","Tecnecio",99,
"7/6/4",1.36,0,0,-95.5,1.9
9044 DATA 44,"Ru","Rutenio",101.
07,"3/4/8/6/2",1.34,0,8.3,-145,2
.2
9045 DATA 45,"Rh","Rodio",102.91
,"3",1.34,0,8.3,-162,2.2
9046 DATA 46,"Pd","Paladio",106.
4,"2/4/3",1.37,0,8.9,-98.5,2.2
9047 DATA 47,"Ag","Plata",107.87
,"1/2",1.44,1.53,10.3,-193,1.9
9048 DATA 48,"Cd","Cadmio",112.4
0,"2/1",1.54,1.48,13.1,26.1,1.7
9049 DATA 49,"In","Indio",114.82
,"3",1.66,1.44,15.7,-19.3,1.7
9050 DATA 50,"Sn","Estano",118.6
9,"4/2",1.62,1.41,16.3,-99.5,1.8
9051 DATA 51,"Sb","Antimonio",12
1.75,"+-3/5",1.59,1.38,18.4,-90.
5,1.9
9052 DATA 52,"Te","Teluro",127.6
0,"4/6/-2",1.60,1.35,20.5,-189,2
.1
9053 DATA 53,"I","Iodo",126.90,"
+-1/5/7",0,1.33,25.7,-295.2,2.5
9054 DATA 54,"Xe","Xenon",131.30
,"0",0,2.09,42.9,43.5,0
9055 DATA 55,"Cs","Cesio",132.91
,"1",2.67,2.25,70,-36.7,0.7
9056 DATA 56,"Ba","Bario",137.34
,"2",2.22,1.98,39,46.4,0.9
9057 DATA 57,"La","Lantano",138.
91,"3",1.87,1.69,22.5,-53.1,1.1
9058 DATA 58,"Ce","Cerio",140.12
,"3/4",1.81,1.65,21.0,0,1.1
9059 DATA 59,"Pr","Praseodimio",
140.91,"3/4",1.82,1.65,20.8,0,1.
1

9060 DATA 60,"Nd","Neodimio",144
.24,"3",1.82,1.64,20.6,0,1.2
9061 DATA 61,"Pm","Promecio",147
,"3",0,0,0,0,0
9062 DATA 62,"Sm","Samario",150.
35,"3/2",1.81,1.66,19.9,0,1.2
9063 DATA 63,"Eu","Europio",151.
96,"3/2",1.99,1.85,28.9,0,0
9064 DATA 64,"Gd","Gadolinio",15
7.25,"3",1.79,1.61,19.9,0,1.1
9065 DATA 65,"Tb","Terbio",158.9
2,"3/4",1.80,1.59,19.2,0,1.2
9066 DATA 66,"Dy","Disprosio",16
2.50,"3",1.80,1.59,19.0,0,0
9067 DATA 67,"Ho","Holmio",164.9
3,"3",1.79,1.58,18.7,0,1.2
9068 DATA 68,"Er","Erbio",167.26
,"3",1.78,1.57,18.4,0,1.2
9069 DATA 69,"Tm","Tulio",168.93
,"3",1.77,1.56,18.1,0,1.2

9070 DATA 70,"Yb","Yterbio",173.
04,"3/2",1.94,1.70,24.8,0,1.1
9071 DATA 71,"Lu","Lutecio",174.
97,"3",1.75,1.56,17.8,0,1.2
9072 DATA 72,"Hf","Hafnio",178.4
9,"4/3/2",1.67,0,13.6,60.8,1.3
9073 DATA 73,"Ta","Tantalio",180
.95,"5",1.49,0,10.2,-14.4,1.5
9074 DATA 74,"W","Wolframio",183

.85,"6/5/4/2",1.41,0,9.53,-119,1
.7
9075 DATA 75,"Re","Renio",186.2,
"7/6/4",1.37,0,8.85,-36.7,1.9
9076 DATA 76,"Os","Osmio",190.2,
"4/8/6/3/2",1.35,0,8.43,-139,2.2
9077 DATA 77,"Ir","Iridio",192.2
,"3/4",1.36,0,8.54,-190,2.2

9078 DATA 78,"Pt","Platino",195.
09,"4/2/3",1.39,0,9.10,-247,2.2
9079 DATA 79,"Au","Oro",196.97,"
3/1",1.46,1.50,10.2,-270,2.4
```

GUSANEZ

por José C. Tomás







```
9080 DATA 80,"Hg","Mercurio",200
.59,"2/1",1.57,1.49,14.8,18.6,1.
9
9081 DATA 81,"Tl","Talio",204.37
,"3",1.71,1.48,17.2,30.4,1.8

9082 DATA 82,"Pb","Plomo",207.19
,"2/4",1.75,1.47,18.3,-99.5,1.8

9083 DATA 83,"Bi","Bismuto",208.
98,"3/5",1.70,1.46,21.3,-91.5,1.
9
9084 DATA 84,"Po","Polonio",210,
"2/3/4/6/2",1.76,0,22.7,-127,2

9085 DATA 85,"At","Astato",210,"
5/3/1",0,0,0,-270,2.2
9086 DATA 86,"Rn","Radon",222,"0
",0,2.14,0,0,0
9087 DATA 87,"Fr","Francio",223,
"1",0,0,0,0,0.7

9088 DATA 88,"Ra","Radio",226,"2
",0,0,45,0,0.9
9089 DATA 89,"Ac","Actinio",227,
"3",1.88,0,0,0,1.1
9090 DATA 90,"Th","Torio",232.04
,"4/3/2",1.80,1.65,19.9,0,1.3

9091 DATA 91,"Pa","Protoactinio"
,231,"5/4",1.61,0,15.0,0,1.5
9092 DATA 92,"U","Uranio",238.04
,"6/4/3/5",1.38,1.42,12.5,0,1.7

9093 DATA 93,"Np","Neptuno",237,
"5/4/6/3",1.3,0,21.1,0,1.3
9094 DATA 94,"Pu","Plutonio",242
,"4/6/3/5",1.51,0,0,0,0

9095 DATA 95,"Am","Americio",243
,"3/4/5/6",1.73,0,0,0,0
9096 DATA 96,"Cm","Curio",247,"3
",0,0,0,0,0

9097 DATA 97,"Bk","Berkelio",247
,"3/4",0,0,0,0,0
9098 DATA 98,"Cf","Californio",2
51,"3",0,0,0,0,0
9099 DATA 99,"Es","Einstenio",25
4,"3",0,0,0,0,0

9100 DATA 100,"Fm","Fermio",253,
"3",0,0,0,0,0
9101 DATA 101,"Md","Mendelevio",
256,"3",0,0,0,0,0

9102 DATA 102,"No","Nobelio",254
,"3/2",0,0,0,0,0
9103 DATA 103,"Lw","Laurencio",2
57,"-",0,0,0,0,0

9104 DATA 104,"Ku","Kurchatovio"
,0,"-",0,0,0,0,0
9105 DATA 105,"Ha","Hahnio",0,"-
",0,0,0,0,0
```

MASTERMIND

La versión de este conocido juego realizada por Juan I. Perea nos da únicamente seis intentos para acertar la combinación correcta. Pero si la paciencia no es vuestro fuerte, podéis alterar el programa para que admita un máximo de oportunidades.

Dispone de opciones para televisor en color o en blanco y negro y la presentación es inmejorable. Además, para que todos os decidáis a teclearlo, el listado es considerablemente breve.

```
   1 REM MasterMIND 1985 Perea
 Software
  10 LET bn=2: RESTORE : FOR f=U
SR "a" TO USR "t"+7: READ g: POK
E f,g: NEXT f: DIM p(4,2): FOR f
=1 TO 4: READ p(f,2),p(f,1): NEX
T f
  20 DATA 0,g,g,g,255,g,g,g,16,8
,4,2,255,g,g,g,128,192,160,144,1
43,135,131,129,4,2,1,g,255,g,254
,252
  30 DATA 128,g,g,g,255,g,127,62
,130,193,161,145,137,133,3,1,1,g
,g,g,255,g,g,g,192,48,12,2,255,g
,g,g
  40 DATA 32,64,128,g,255,g,127,
62,255,129,g,g,g,g,1,g,63,66,129
,g,255,g,126,60,252,2,1,g,g,g,2,
4
  50 DATA 0,7,31,63,g,127,g,g,0,
224,248,252,220,238,g,g,127,g,g,
59,60,31,7,0,238,g,g,220,60,248,
224,0
  60 DATA 0,7,24,32,g,64,g,g,0,2
24,24,4,g,2,g,g,64,g,g,32,g,24,7
,0,2,g,g,4,g,24,224,0
  90 DATA -4,4,-4,6,-2,4,-2,6
 100 BORDER 1: BRIGHT 0: INVERSE
 0: OVER 0: FLASH 0: PAPER 1: IN
K 7: CLS : PRINT  PAPER 2; INK 7
-(5 AND bn<2); FLASH 1;"
PARA EL CASSETTE           ": DIM s
$(32)
 110 PRINT AT 10,5;"MN QR  MASTE
R   QR MN              OP ST    MI
ND  ST OP";AT 13,5; 1985 Perea
 Software";AT 21,0; PAPER 2;s$
 120 FOR f=1 TO 4: PLOT 40-5*f,6
4-5*f: DRAW 168+10*f,0: DRAW 0,4
0+10*f: DRAW -168-10*f,0: DRAW 0
,-40-10*f: NEXT f
 125 IF bn=2 THEN  INPUT "Blanco
 y negro ?  "; LINE a$: LET a$=(
a$+" ")(1): LET bn=a$="s" OR a$=
"S"
 130 IF INKEY$="" THEN  GO TO 13
0
 140 CLS : DIM n$(4): FOR f=1 TO
 4
 150 LET n$(f)=STR$ (1+INT (RND*
6)): FOR g=1 TO f-1: IF n$(f)=n$
(g) THEN  GO TO 150
 160 NEXT g: NEXT f
 200 BORDER 4: PAPER 4: INK 0: C
LS : PAPER 7: FOR f=3 TO 19: PRI
NT AT f,0;s$;AT f,31;"mini MASTE
R MIND "(f-2): NEXT f
 210 PRINT AT 9,2;"AB   CD   EF
  GH   IJ  KL": DIM f$(6,4): LE
T n=0
1000 LET n=n+1: IF n>6 THEN  GO
TO 7000
1010 PLOT n*40-36,20: DRAW 39,0:
 DRAW 0,127: DRAW -39,0: DRAW 0,
-127
1020 LET cl=n*5-3: LET h=0: LET
i=h: FOR f=11 TO 17 STEP 2: PRIN
T AT f,cl;"QR";AT f+1,cl;"ST": N
EXT f
1030 LET h=h+1: IF h=5 THEN  LET
 h=1: LET i=1
1040 LET ln=9+2*h
1050 PRINT  INK 8; OVER 1;AT ln,
cl;"QR";AT ln+1,cl;"ST"
1055 LET a$=INKEY$: IF a$=CHR$ 1
3 THEN  GO TO 1300
1060 IF a$<"1" OR a$>"6" THEN  F
OR f=1 TO 5: NEXT f: GO TO 1050
1070 BEEP .1,30: LET f$(n,h)=a$:
 INK VAL a$: GO SUB 5000
1080 IF INKEY$<>a$ THEN  GO TO 1
030
1090 GO TO 1080
1300 IF NOT i THEN  FOR f=1 TO 4
: NEXT f: GO TO 1050
1310 FOR f=1 TO 3: FOR g=f+1 TO
4: IF f$(n,f)=f$(n,g) THEN  GO T
O 1055
1320 NEXT g: NEXT f: INK 8: GO S
UB 5000
1330 BEEP .1,-20: BEEP .2,10: LE
T ng=0: LET bc=ng: FOR f=1 TO 4:
 IF n$(f)=f$(n,f) THEN  LET ng=n
g+1: NEXT f: GO TO 1360
1340 FOR g=1 TO 4: IF n$(f)=f$(n
,g) THEN  LET bc=bc+1
1350 NEXT g: NEXT f
1360 DIM h(4): FOR f=1 TO ng
1370 LET g=INT (RND*4)+1: IF h(g
) THEN  GO TO 1370
1380 LET h(g)=2: PRINT AT p(g,1)
,5*n+p(g,2);"MN";AT p(g,1)+1,5*n
+p(g,2);"OP": NEXT f
1400 FOR f=1 TO bc
1410 LET g=INT (RND*4)+1: IF h(g
) THEN  GO TO 1410
1420 LET h(g)=1: PRINT AT p(g,1)
,5*n+p(g,2);"QR";AT p(g,1)+1,5*n
+p(g,2);"ST": NEXT f
1500 IF ng<4 THEN  INVERSE 1: PL
OT n*40-36,20: DRAW 39,0: DRAW 0
,127: DRAW -39,0: DRAW 0,-127: I
NVERSE 0: GO TO 1000
2000 GO TO 7000
5000 LET x=8*cl+3: LET y=166-8*l
n: PRINT AT ln,cl;"MN";AT ln+1,c
l;"OP": IF bn THEN  INVERSE 1: G
O SUB 5943+PEEK (22528+cl+32*ln)
: INK 0: INVERSE 0
5100 RETURN
6000 PLOT x+3,y: DRAW 0,6: RETUR
N
6001 PLOT x+3,y: DRAW -3,0: DRAW
 0,3: DRAW 3,0: DRAW 0,3: DRAW -
3,0: RETURN
6002 PLOT x,y: DRAW 3,0: GO SUB
6000: DRAW -3,0: PLOT x+3,y+3: D
RAW -3,0: RETURN
6003 GO SUB 6000: PLOT x,y+6: DR
AW 0,-3: DRAW 3,0: RETURN
```

```
6004 PLOT x,y: DRAW 3,0: DRAW 0,
3: DRAW -3,0: DRAW 0,3: DRAW 3,0
: RETURN
6005 PLOT x,y: DRAW 0,3: GO TO 6
004
7000 IF n<7 THEN  GO TO 8000
7010 PRINT AT 0,0;s$;s$;AT 0,0;"
La combinacion era :": FOR f=1 T
O 4: LET ln=0: LET cl=f*2+21: IN
K VAL n$(f): GO SUB 5000: NEXT f
7020 GO TO 8030
8000 PRINT AT 0,0;"Acertaste la
combinacion en     ";n;" intento
";("s" AND n>1);s$( TO 22+(n=1))
8030 RESTORE 9000: FOR f=1 TO n:
 READ a$: NEXT f: PAPER 4: PRINT
 AT 21,0;a$: BEEP 1,0: FOR f=1 T
O 7: BEEP .03,20: NEXT f
8040 IF INKEY$<>"" THEN  GO TO 8
040
8050 PAUSE 1: PAUSE 0: GO TO 100
9000 DATA "Has tenido mucha suer
te","Eso es suerte","Todo un mae
stro","Buena vista","Psch... Del
 monton","Muy apurado","Lo tuyo
no es esto"
9990 CLEAR : SAVE "MasterMIND" L
INE 10
9999 BRIGHT 0: BORDER 7: PAPER 7
: INK 0: CLS
```



DINAMICA

Utilizando el desarrollo en serie de Taylor, el programa calcula la posición de un móvil cuyo movimiento está regido por ecuaciones más o menos complejas. Tras la presentación e instrucciones solicita la introducción de la expresión de la aceleración en cada una de sus componentes espaciales (ax, ay, az). La estructura del programa está claramente expuesta en el mismo por medio de sentencias REM.

Juan Carlos Fabero

```
  9 REM ENTRADA DE DATOS
 10 PRINT AT 20,0;" Introduzca
las aceleraciones.  "
 15 INPUT "Ax="; LINE a$: INPUT
 "Ay="; LINE b$: INPUT "Az="; LI
NE c$
 17 IF a$="" OR b$="" OR c$=""
THEN  BEEP .01,30: GO TO 10
 20 PRINT AT 20,0;"Introduzca l
a posicion inicial. "
 25 INPUT "x=";x: INPUT "y=";y:
 INPUT "z=";z
 30 PRINT AT 20,0;"Introduzca l
a velocidad inicial."
 35 INPUT "Vx=";u: INPUT "Vy=";
v: INPUT "Vz=";w
 40 PRINT AT 20,0;"Introduzca e
l incremento h.          "
 45 INPUT "h=";h
 50 PRINT AT 20,0;"Introduzca e
l intervalo de        presentacion
 de resultados."
 55 INPUT "Intervalo=";int
 60 PRINT AT 20,0;"Desea mensaj
e de scroll? (s/n). ",,
 65 LET r$=INKEY$: IF r$="" THE
N  GO TO 65
 70 IF r$="s" OR r$="S" THEN  L
ET sc=1: GO TO 85
 75 IF r$<>"n" AND r$<>"N" THEN
 GO TO 65
 80 LET sc=0
 85 CLS
 90 LET t=0
100 REM FUNCIONES
110 DEF FN a(u,v,w)=VAL a$
120 DEF FN b(u,v,w)=VAL b$
130 DEF FN c(u,v,w)=VAL c$
135 REM DERIVADAS
140 DEF FN p(u,v,w,h)=(FN a(u-2
*h,v,w)-8*FN a(u-h,v,w)+8*FN a(u
+h,v,w)-FN a(u+2*h,v,w))/12/h
150 DEF FN q(u,v,w,h)=(FN b(u,-
2*h+v,w)-8*FN b(u,-h+v,w)+8*FN b
(u,h+v,w)-FN b(u,2*h+v,w))/12/h
160 DEF FN r(u,v,w,h)=(FN c(u,v
,w-2*h)-8*FN c(u,v,w-h)+8*FN c(u
,v,w+h)-FN c(u,v,w+2*h))/12/h

165 REM CALCULO SERIE DE TAYLOR
170 LET k=0
180 LET k=k+1
185 IF INKEY$=CHR$ 13 THEN  GO
TO 600
190 LET t=t+h
200 LET p=FN p(u,v,w,1e-6): LET
 q=FN q(u,v,w,1e-6): LET r=FN r(
u,v,w,1e-6)
210 LET x=x+u*h+h*h*FN a(u,v,w)

/2+h*h*h*p/6
220 LET y=y+v*h+h*h*FN b(u,v,w)
/2+h*h*h*q/6
230 LET z=z+w*h+h*h*FN c(u,v,w)
/2+h*h*h*r/6
240 LET u=u+h*FN a(u,v,w)+h*h*p
/2
250 LET v=v+h*FN b(u,v,w)+h*h*q
/2
260 LET w=w+h*FN c(u,v,w)+h*h*r
/2
270 IF k<int/h THEN  GO TO 180
275 REM PRESENTACION RESULTADOS
280 PRINT "t=";t'"x=";x'"y=";y'
"z=";z
290 PRINT "--------------------
------------"

295 POKE 23692,PEEK 23692*sc
300 GO TO 170
340 REM INSTRUCCIONES
350 CLS
360 CLS : PRINT " Este programa
 se sirve del de- sarrollo en se
rie de Taylor para"
370 PRINT "calcular la posicion
 de un movilque se mueve bajo lo
s efectos deuna determinada fuer
za (o fuer- zas)."
380 PRINT " Solo debe introduci
r las formu-las de las aceleraci
ones que su-fre el movil en cada
 uno de los ejes cartesianos."
390 PRINT '" Las formulas se al
macenan como cadenas alfanumeric
```

```
as,por lo queno deben contener n
ingun error"
 400 PRINT "sintactico o se forz
ara un men- saje  C NONSENSE IN
BASIC."
 410 PRINT " Por la misma razon,
 cuando la  fuerza en una de las
 tres direc-ciones sea nula, deb
e introducir""0"" y no simplemen
te la cadena  vacia."
 420 PRINT )1;AT 1,8;"PULSE UNA
TECLA"
 430 FOR n=0 TO 10000: IF INKEY$
="" THEN  NEXT n
 440 CLS
 450 PRINT " El programa le pedi
ra el valor del incremento de ti
empo para laserie de Taylor."
 460 PRINT " Cuanto mas pequeno
sea h, mas  preciso y lento sera
 el calculo."
 470 PRINT '" Tambien se le pedi
ra cada cuan-tas unidades de tie
mpo quiere   presentar los resul
tados."
 480 PRINT ''"  Para detener el
calculo pulse              'ENTER'
"

 530 PRINT )1;AT 1,8;"PULSE UNA
TECLA"
 540 FOR n=0 TO 2000: IF INKEY$=
"" THEN  NEXT n
 545 REM dibujo de los ejes
 550 CLS : PLOT 127,87: DRAW 0,8
7
 560 PLOT 127,87: DRAW 87,0
 570 PLOT 127,87: DRAW -49,-49
 580 PRINT AT 0,15;"Z";AT 10,28;
"Y";AT 17,8;"X"
 590 GO TO 10
 600 REM   STOP
 610 PRINT )1;AT 0,0;" Pulse SPA
CE para continuar o   'P' para f
inalizar."
```

```
z =80
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
t =2
x =86
y =168
z =220
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
t =3
x =134
y =277
z =420
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
t =4
x =192
y =416
z =680
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
t =5
x =260
y =585
z =1000
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
 620 LET r$=INKEY$: IF r$="" THE
N  GO TO 620
 630 IF r$=CHR$ 32 THEN  PRINT )
1;AT 0,0,,,,: GO TO 190
 640 IF r$<>"p" AND r$<>"P" THEN
 GO TO 620
 650 STOP : GO TO 545
 700 REM PARADA DE CINTA
 710 BEEP .3,30
 720 PRINT AT 10,8; INK 0; PAPER
 7; FLASH 1;"PARE LA CINTA"
 730 FOR n=0 TO 400: IF INKEY$="
" THEN  NEXT n
 740 PAPER 0: INK 7: BRIGHT 0: B
ORDER 0: CLS
 750 PRINT "********************
************"
 760 FOR n=1 TO 18: PRINT AT n,0
;"*
 *": NEXT n
 770 PRINT "********************
************"
 780 PRINT AT 4,0;"*   RESOLUCIO
N DE PROBLEMAS    *"
 790 PRINT AT 8,0;"*           DI
NAMICOS           *"
 800 PRINT AT 12,0;"*  Juan Carl
os Fabero Jimenez  *";AT 14,13;"
1986"
 810 PRINT AT 21,8;"PULSE UNA TE
CLA"
 820 FOR n=0 TO 400: IF INKEY$="
" THEN  NEXT n
 830 GO TO 360
9999 CLEAR : SAVE "DINAMICA" LIN
E 700
```

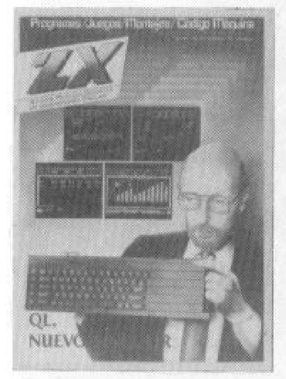


SERVICIO DE

Completa tu colección de ZX.
A continuación te resumimos el contenido de los ejemplares atrasados en existencia.



Núm. 3/300 ptas.
El Spectrum por dentro. Quince programas, juegos y montajes Software.



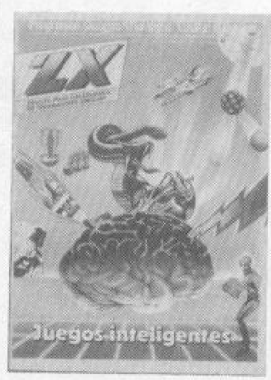
Núm. 4/300 ptas.
QL, el nuevo Sinclair. Dieciocho programas, juegos, montajes, ideas/Novedades.



Núm. 5/300 ptas.
Gráficos y sonido en el Spectrum/Libros/Software/13 programas.



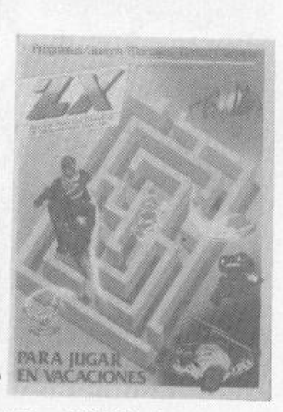
Núm. 6/300 ptas.
Construya su propio juego/13 programas y montajes/ideas/Software.



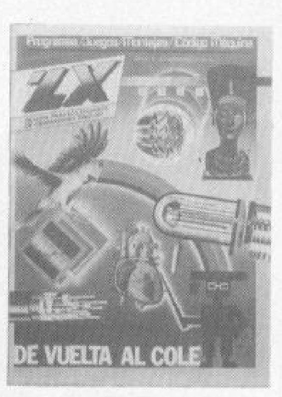
Núm. 7/300 ptas.
Juegos inteligentes/Software/ 11 programas/Libros.



Núm. 8/300 ptas.
La aventura es la aventura/12 programas/Juegos y montajes/Código máquina.



Núm. 9/300 ptas.
Construye tu propio juego. Catorce programas para el verano. Gráficos en el Spectrum.



Núm. 10/300 ptas.
Catorce programas educativos: geografía, cramer, gráficos, razones trigonométricas, elongación. Código máquina.



Núm. 11/300 ptas.
Cómo crear marcianos y otros monstruos.

Diez programas satélites de júpiter, rescate, interés, círculo, préstamo hipotecario.



Núm. 12/300 ptas.
Presentación del Spectrum Plus. Forth, capítulo 1. Gráficos en el Spectrum, 4 parte. Libros. Programas y montajes.



Núm. 13/300 ptas.
Guía del software para el Spectrum todos los programas del mercado. Forth, capítulo 2. Visitamos Sinclair Research. Libros. Programas.



Núm. 14/300 ptas.
Cómo jugar al Hobbit. Gráficos de funciones. Programas de ajedrez. Conexiones con el P I/O. Programas Multiplic, enseñar deleitando. Libros, Forth, tercera parte.



Núm. 15/300 ptas.
Simuladores de vuelo. Forth, cuarta parte. Montajes: Reloj digital para Spectrum. BASIC para principiantes. Libros. Programas.

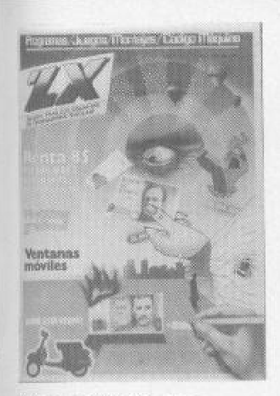


Núm. 16/300 ptas.
Cassettes: solución a los problemas de grabación. Test de Psicología. Sistema de Desarrollo para el ZX-81. Cinemática. Programas. Animación Gráfica. BASIC para principiantes (2). Forth, quinta parte.



Núm. 17/300 ptas.
Mapa de Atic-Atac. Estira de caracteres. Dinámica de una partícula. Libros. QL Magazine. Programas. Convertidor analógico-digital con el P I/O.

EJEMPLARES ATRASADOS



Núm. 18/300 ptas.
Rentas 85. Forth, sexta parte. Programas. BASIC para principiantes (3). Plotting Gráficos. Libros. Usuarios. Crítica.



Núm. 19/300 ptas.
Mapa de Knight Lore. Noticias. Crítica. Renta 85 (segunda parte). Libros. El ZX-81 aprende a sumar. Scroll de ventanas. Programas. El software que nos invade. BASIC para principiantes (4).



Núm. 20/300 ptas.
Vacaciones con informática. Crítica. Noticias. Programas. Son muy divertidos. Libros. Generación de placas de circuito impreso. Forth. Movimiento armónico simple. Spectrum musical.



Núm. 21/300 ptas.
Mapa de Underwurlde. Noticias. Crítica. ¿Has probado? Programa especial: barquitos. Sois muy divertidos. Libros para el verano. Un poco de física. BASIC para principiantes (5).



Núm. 22/300 ptas.
Noticias. Teclados profesionales. Crítica. ¿Has probado? Programa especial: procesador de textos. Generación de placas de circuito impreso (segunda parte). Programas QL español. Quinielas en Spectrum. BASIC para principiantes (6).



Núm. 23/300 ptas.
Crítica. ¿Has probado? Profanation profanado. Noticias. Discos para Spectrum. Dossier educación: Spectrum en el aula, autoevaluación, Logo. Código máquina. Programación especial: quinielas. Montaje a cámara lenta. BASIC para principiantes (7).



Núm. 24/300 ptas.
Juegos/Mapas del Nodes of Yesod y Lords of Midnight/¿Has probado?/Sois muy divertidos/Usuario/Ajuste de gráficas/Multisearch/Programas/Montaje: inversor de video para ZX 81/Dossier QL.



Núm. 25/300 ptas.
Juegos/Especial juegos, Mapas y trucos de: Highway encounter, Tir Na Nog, Nightshade/¿Qué es el Stack?/Programa especial/Código máquina/Lotería primitiva/Stándares de la informática/Programas.

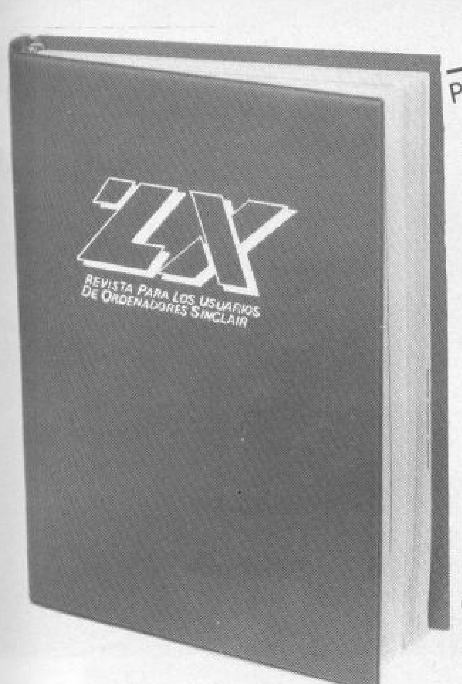


Núm. 26/300 ptas.
Spectrum o QL, invasión de los 128/¿Cómo utilizar mejor el microdrive?/Juegos/Mapa del Dun Darach y misión imposible/Programación estructurada/BASIC.



Núm. 27/300 ptas.
La vida de Sinclair/Piezas musicales para Spectrum/Juegos/Mapas del ARNHCM y SABOTEUR/Areas/BASIC para impresora/El área de variable y la instrucción RST 16.

DISPONEMOS DE TAPAS ESPECIALES PARA SUS EJEMPLARES DE ZX (sin necesidad de encuadernación)



PRECIO UNIDAD **650** ptas.

(en cada tomo se pueden encuadernar 6 números)

Para hacer tu pedido, rellena el cupón adjunto, córtalo y envíalo HOY MISMO a:

ZX, Bravo Murillo, 377 • 28020-MADRID • Tel. 733 74 13

Los ejemplares atrasados de ZX serán una fuente constante de conocimientos, ideas, soluciones y entretenimientos para el futuro. Todo lo anterior hace recomendable que los guardes ordenadamente en una de las tapas especiales para ZX. Cada tapa puede contener 6 ejemplares y cuesta solamente 650 ptas.

Ruego me envien los siguientes ejemplares atrasados de ZX .... al precio de 300 ptas. cada uno

Por favor envien .... tapa(s) al precio de 650 ptas. cada una (+ gastos de envio).

El importe lo abonaré:
□ contra reembolso □ cheque adjunto □ con mi tarjeta de crédito
□ American Expres □ Visa □ Interbank.

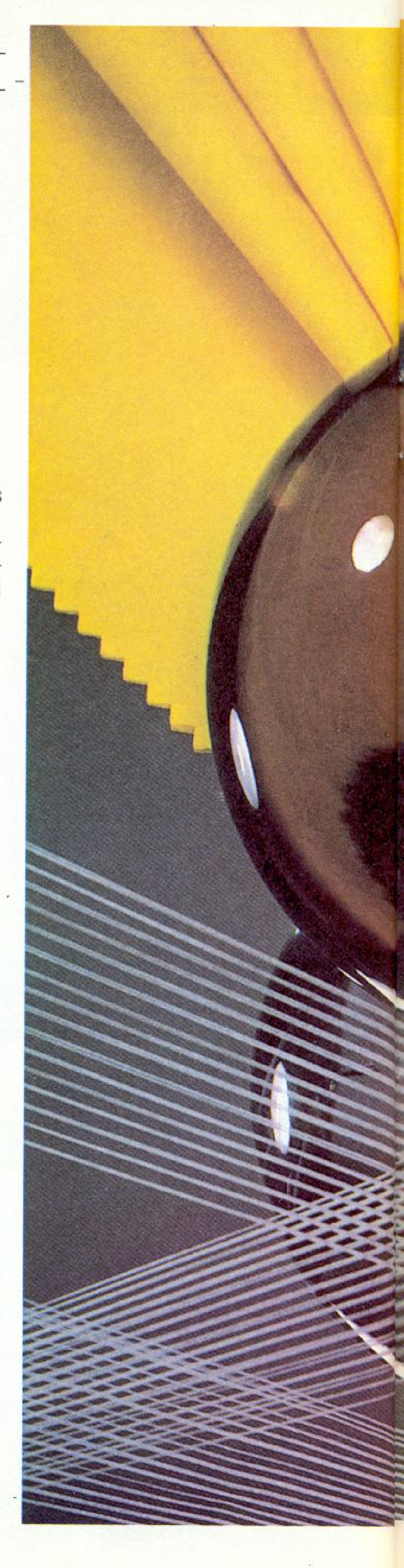
Fecha de caducidad ....

Número de mi tarjeta [ ]

NOMBRE ....
DIRECCION ....
POBLACION .... C.P. ....
PROVINCIA ....

SUCESIONES

Dentro del enorme mundo de las matemáticas encontramos una pequeña parcela dedicada a las sucesiones. En este artículo se va a tratar precisamente este tema, enfocado desde el punto de vista de la programación.



LAS sucesiones son conjuntos de números reales, ordenados y que, por lo general, obedecen una ley, relacionando el valor del elemento con el lugar que ocupan dentro de la sucesión. Esta regla o ley es una expresión matemática en la que «n» es un número natural.

Uno de los conceptos más importantes de las sucesiones es el de límite o punto de los reales en el cuál se cumple que la diferencia entre él y cualquier término lo suficientemente avanzado es tan pequeña o más que épsilon (siendo épsilon un número positivo tan pequeño como queramos). En caso de que dos o más puntos cumplan esta condición no se llamarán límites, sino puntos de acumulación.

Otros conceptos con los que trabajaremos son los de cota, decreciente, creciente, convergente, oscilante,...

Problemas más importantes

Al estudiar a fondo este tema encontramos varios problemas, debidos a que nuestro micro, que lo sabe casi todo, no trabaja con números con infinito o épsilon. Tampoco calcula algunas operaciones comúnmente usadas en matemáticas como es el caso de la potencia de un número negativo (−y ↑ x).

La solución al primer problema es bastante sencilla, aunque tiene sus limitaciones. Consiste en sustituir infinito por un número que cumpla que si le sumamos un real no demasiado grande, ya que trabajaremos con cantidades no excesivas, nos de él mismo. Además, cuando lo multipliquemos por cualquier número, éste se almacena, por si queremos reducir o simplificar.

Para buscar un número que cumpla estas condiciones debemos recurrir a números con exponencial (como 1E10):

I+1=I

1E10+1=1E10

(I=infinito)

2I/3I=2/3

2 1E10/3.1E10=2E10/3E10=2/3

Ahora bien, tiene sus defectos:

I+1−I=1

1E10+1−1E10=1E10−1E10=0

Aún así, lo podemos usar, ya que en lo que nos ocupa ahora no nos va a limitar demasiado.

Con épsilon ocurre lo mismo, usaremos 1E−S.

No nos va a resultar tan fácil enseñar al ordendor a operar potencias de base negativa, aunque, como es muy «inteligente», aprenderá rápido.

Observemos:

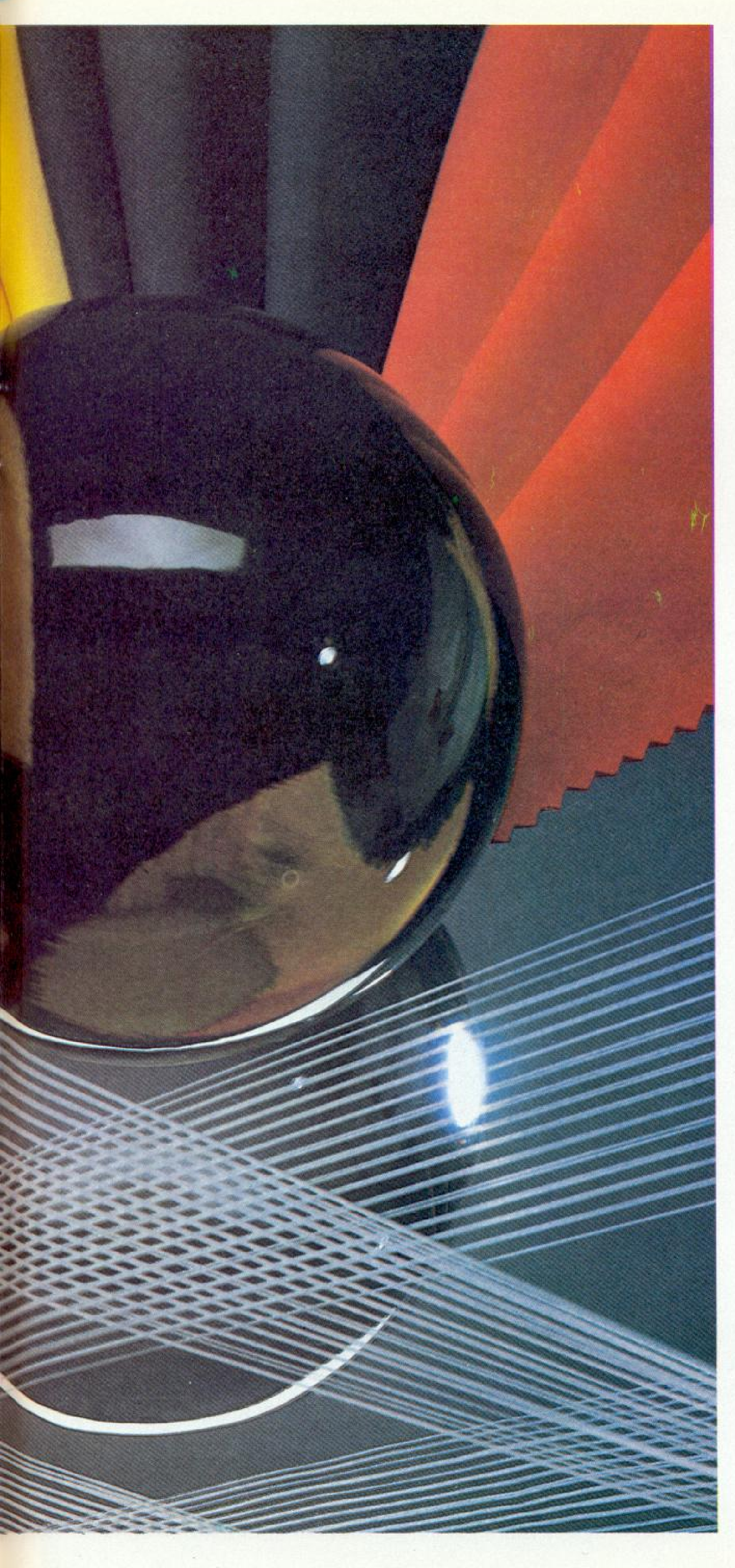
(−1) ↑ 7=−1	(−1) ↑ 10=1
(−2) ↑ 3=−8	(−2) ↑ 2=4

De aquí podemos deducir que si el exponente es impar (columna izquierda), el resultado es siempre negativo y si es par, positivo (siempre trabajando con bases negativas y exponentes enteros). Ya hemos hallado el signo, lo que queda es tan sólo la misma potencia con base positiva.

Teniendo en cuenta estos factores podemos decir:

(−y) ↑ x=y ↑ x −2y ↑ x si x es impar

Esto se entiende ya que



y ↑ x–2y ↑ x=–y ↑ x, es decir, si x esd impar el signo será negativo.

Ahora debemos buscar una fórmula que dé 1 si x es impar ó 0 si x es par, para que, multiplicando con –2y ↑ x, la resta se lleva a cabo o no, y por lo tanto se cambie de signo si es necesario.

Esa fórmula es:

INT x+1 –INT x 2 2

Si x es impar, y por supuesto entero, el primer término será una unidad mayor que el segundo. Por lo tanto su diferencia es uno. Si x es par ambos serán iguales y el resultado será 0 (como buscábamos).

Para poner en práctica esto en el programa, debemos definir una función enn la que se introduzca y,x. Pero encontramos un error, si lo usamos con base positiva nos saldrá mal. Para subsanar este pequeño fallo tenemos que seguir el mismo criterio que antes; si es positivo no se restará (multiplicando por 0), y si es negativo se efectuará la operación (multiplicando por 1).

La fórmula que nos puede ayudar es:

ABS (y)–y
2*ABS (y)

Finalmente nos queda:

FN r(y,x)=ABSy x–ABSy x*((INT((x+1)/2–INT(x/2))*((ABSy–y)/2*ABSy))

El programa

Aclarados estos conceptos, podemos pasar a la parte en la que el protagonista será el Spectrum, tanto de 16 como de 48 K.

Para empezar, apliquemos estas ideas que acabamos de ver, definiendo «in» como infinito (=1E10), y la función r(y,x) como potencia de base positiva o negativa (líneas 80 y 90).

El programa está dividido en partes bien definidas para comprender mejor su funcionamiento:

–Dibujo de la pantalla (líneas 100 y 220 y DATA's 1500).

Se encarga de la representación inicial en nuestro monitor. No tiene nada en especial en cuanto a programación. Su estructura es bastante sencilla utilizando tan sólo un bucle, combinado con sentencias READ.

–Límites (230 a 290) y *puntos de acumulación* (300 a 440).

Lo primero que debe saber esta

rutina es si realmente hay límite, o por el contrario tenemos puntose de acumulación.

Bajo la forma de término a(n), sólo podemos hacer que halle puntos de acumulación usando la fórmula del signo alternante, es decir, elevando un número negativo a una potencia cuyo exponente sea unas veces par, otras impar. Para comprobar si esto ocurre hemos de ver si continuamente va creciendo o decreciendo la sucesión. En caso de que no se cumpla esta condición, podemos decir que tendrá puntos de acumulación, y el programa se desplazará a la línea 300. Esta función la realizan las líneas 240 y 250.

Para hallar el límite, no encontramos más problema que operar con la función a(n), dando a «n» el valor de infinito (línea 270).

Más difícil resulta hallar los puntos de acumulación, ya que si usásemos el mismo sistema que para hallar el límite sólo podríamos hallar el punto de acumulación cuando «n» es par, al ser todos los números exponenciales pares.

Ahora bien, conociendo el punto de acumulación cuando n es igual a infinito, podemos hayar el otro por el método de aproximación por diferencias. Esto es, la diferencia entre un término par (a), y el punto de acumulación par, será igual a la diferencia entre el punto impar, y un término impar (b) cercano al anterior (a):

P. par −f(a)=P. impar −f(b)

(siendo «a» un número par y b=a+1).

De esta fórmula conocemos todo menos el p. impar que despejándolo sería.

P. impar =P. par −f(a) + f(b)

(Líneas 300 a 440).

Dentro de esta rutina, «límite o puntos de acumulación», también se incluye una SUBrutina situada desde la línea 2000 a la 2210. Esta rutina imprime el límite o los puntos de acumulación teniendo en cuenta que:

si num.>1E10=I: todo número mayor que I es igual a I. Por lo tanto se imprimirá I.

si num.<−1E10=−I: se imprimirá −I por la misma circunstancia.

si nu.<1E−8=ep: se imprimirá 0.

−Cotas superiores e inferiores (líneas 500 a 630).

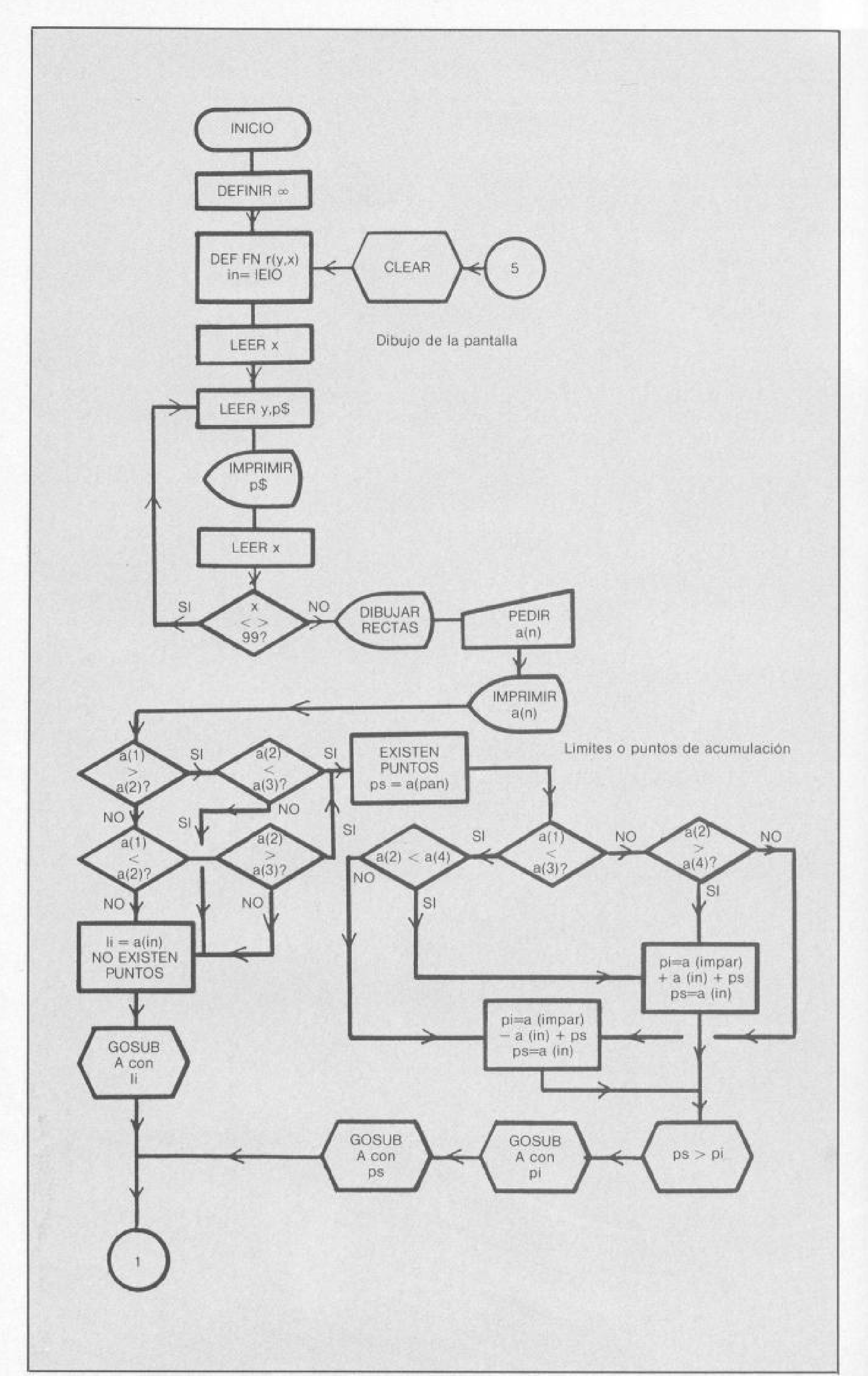
Se denominan cotas superiores, o mayoritarias de una sucesión, a los intervalos que cumplen que todos sus puntos son mayores que los elementos de la sucesión. Llamamos cotas inferiores o minorantes cuando todos los puntos del intervalo son inferiores a los de la sucesión.

Para indicar al ordenador qué puntos son cotas superiores, y cuáles inferiores, debemos fijarnos que: si hay límites, sólo dos puntos, colocados en los estremos de la sucesión. Pueden definir las cotas. El mayor de los dos nos indicará la cota superior, y el menor la inferioir. Estos puntos son el primer término y el límite.

En caso de que haya puntos de acumulación, habrá cuatro puntos que puedan definir las cotas, los dos puntos de acumulación, el primer término y el segundo.

Cuando alguna de las cotas definida por el infinito, noi existirá tal intervalo, por lo que no existirán propiamente cotas.

Para poner en práctica estos conceptos sguiremos unas fórmulas



que son las que usa el programa desde la línea 560 a la 590:

a1>a2, a>a'
sup.: si a>a1 cota igual a
si a<a1 cota igual a1
inf.: si a'<a2 cota igual a'
si a'>a2 cota igual a2

Es estas fórmulas, a1 será f(1) o f(2) según cual sea el mayor, lo mismo ocurre con a2. Los puntos a y a' son los de acumulación.

–Representación gráfica.

Dentro de este bloque encontramos otras subdivisiones: inicialización, representación de enteros, de la sucesión y de a(n).

En el primer apartado, tan sólo se definen ciertas variables, que nos van a ayudar a una mejor gráfica. Estas variables son:

ic = inicio, es el punto que se tomará como referencia para los demás.

px = pixel en el cual se va a representar el punto ic.

pr = proporción que representa cuántos pixel ocupan una unidad en la recta de los reales.

En cuanto a la representación de los enteros, consta de una subrutina (2100) manejada por dos programillas de elemental estructura (800 a 840).

La subrutina es tan sólo una aplicación de, una fórmula que relaciona el número que se va a representar con su situación en la pantalla. Esta fórmula es:

INT((INTic+n–ic)*pr)+px

En donde «n» es un número que va creciendo hasta que no quepa en la pantalla, para luegto empezar a decrecer.

Esta fórmula se explica fácilmente: al número que vamos a representar (INT ic + n) se le resta el punto inicial; esta diferencia, multiplicada por la proporción será los pixel que separan ambos puntos, sumándole el pixel del punto inicial, obtenemos las coordenadas del nuevo punto.

La representación de la sucesión y de a(n) se basan en lo mismo, sólo que, en vez de ser el número a representar un entero, es a(n), donde «n» varía.

–Creciente o decreciente (líneas 100 a 1140).

Una sucesión es creciente, si todos los puntos son mayores o iguales que los que le preceden. Y es decreciente si los elementos van disminuyendo su valor.

Sabiendo esto podemos hacer un muestreo de 5 términos de la sucesión (1010 a 1080), y ver si hay un aumento o una disminución progresiva, comprobando si es monótona, cuando hay varios términos iguales, o estricta si no los hay (1090 a 1140).

–Convergente o divergente.

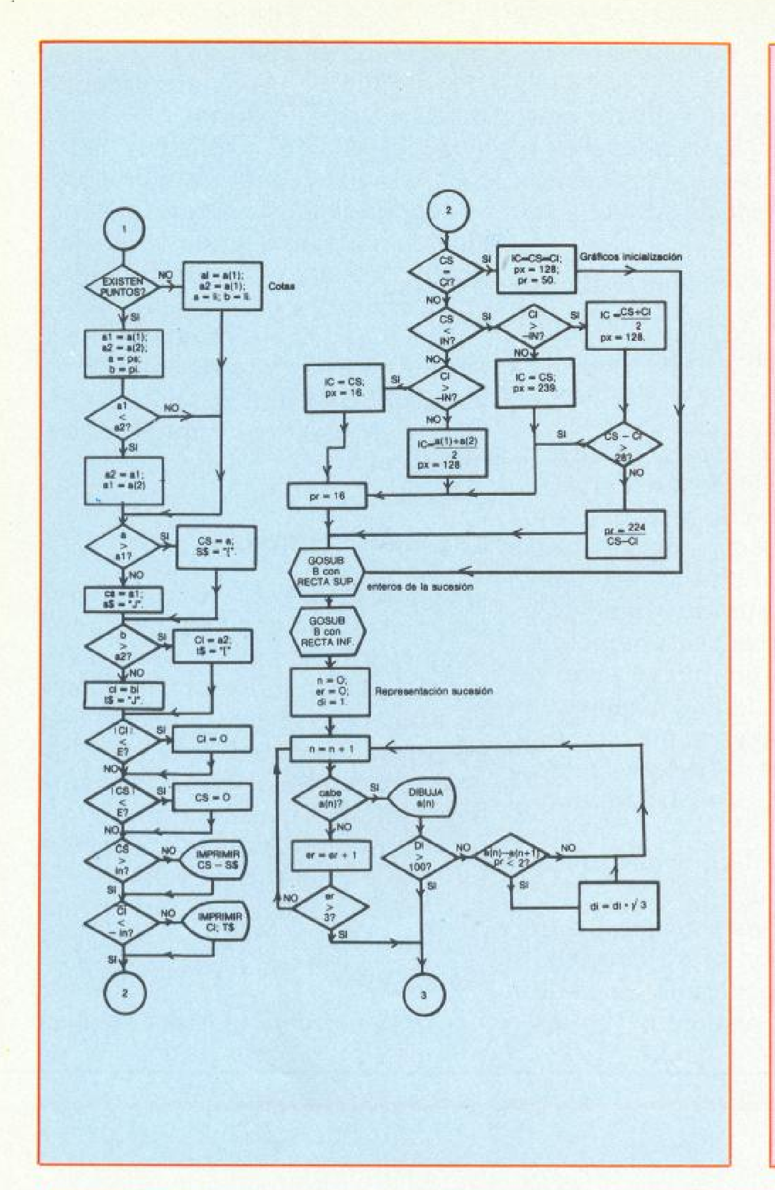
Una progresión es convergente si tiene límite distinto de infinito, es divergente si el límite es infinit, y oscilante cuando tiene puntos de acumulación.

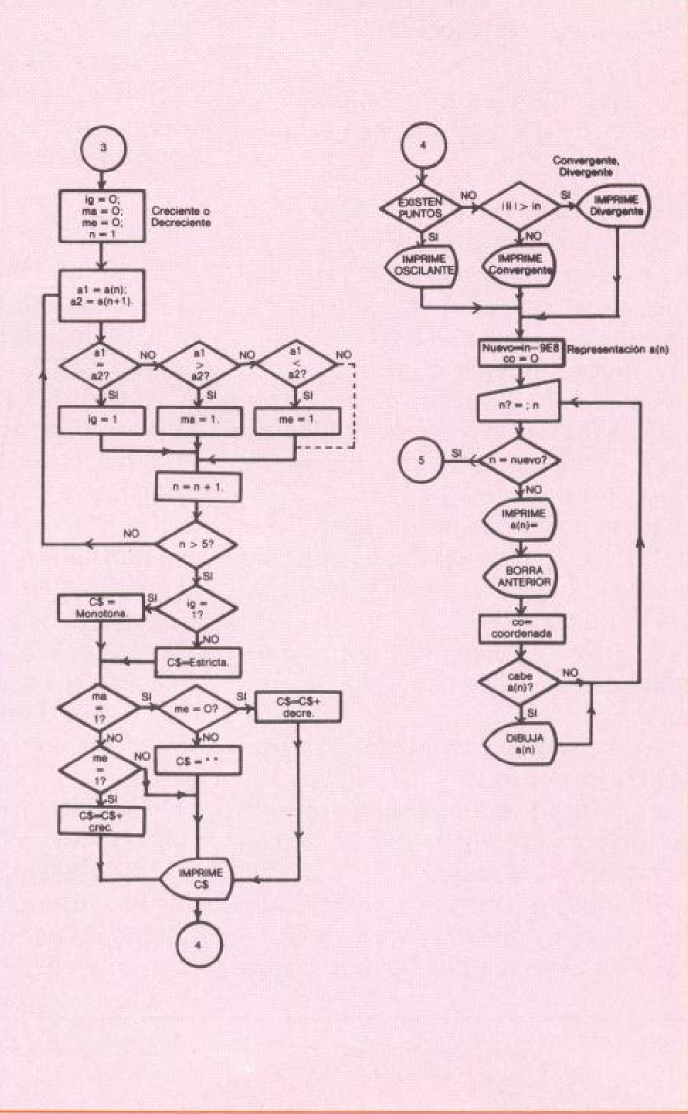
Uso del programa

La utilización de este programa es muy sencilla ya que lo único que se tiene que hacer es introducir al comienzo el término general, teniendo en cuenta que para operar con potencias, puede usar la función FN r(y,x). A continuación el programa hará la representación gráfica y pedirá un número natural para representarlo e imprimir su valor.

Si quiere empezar con nuevos datos debe teclear la palabra «nuevo» y a continuación pulsar ENTER.

Gabriel Ortas González





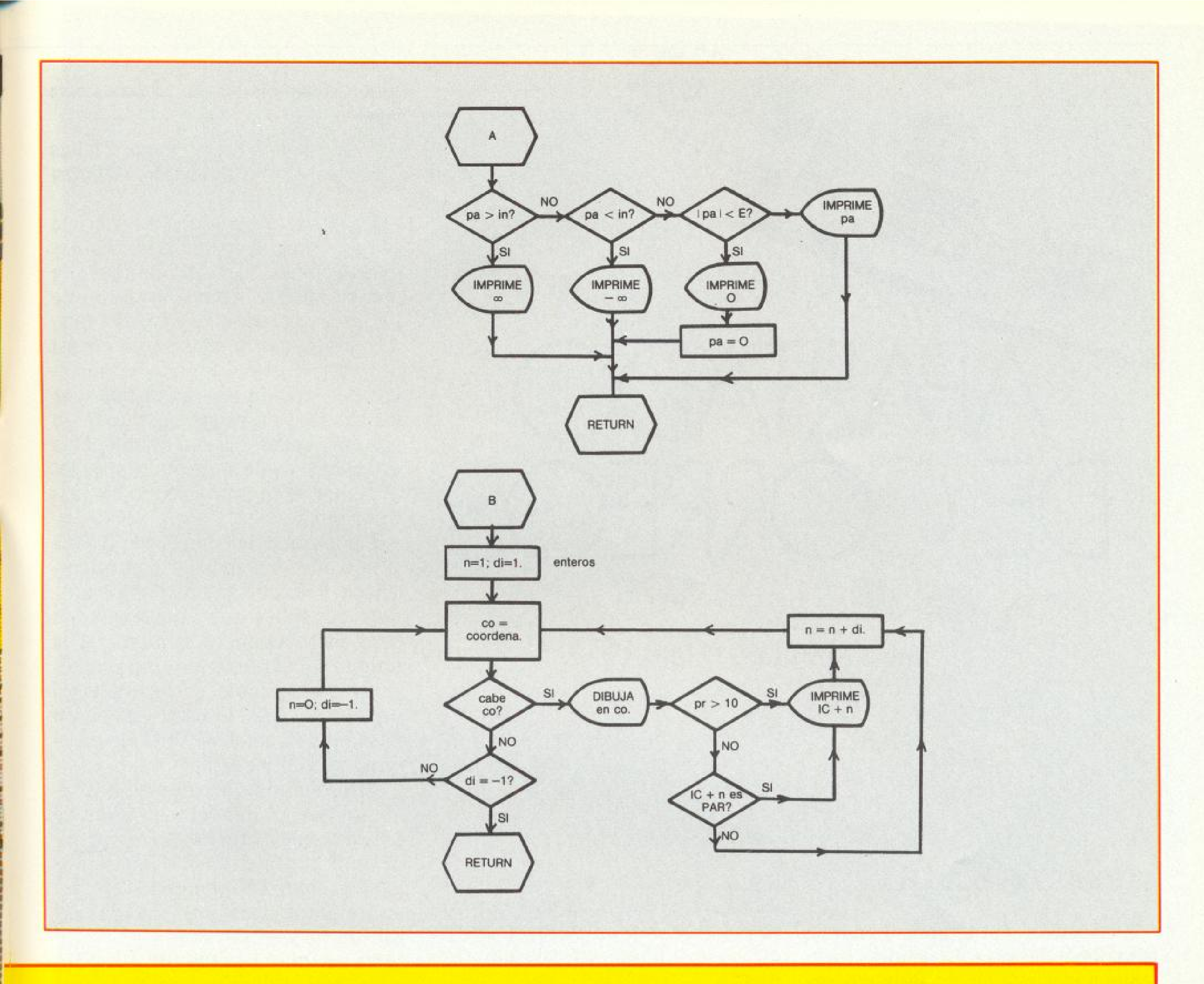

```
  0>REM        SUCESIONES
     GABRIEL ORTAS GONZALEZ
              1985
  50 RESTORE 1530
  60 LET a=0
  70 READ b: POKE USR "a"+a,b: L
ET a=a+1: IF a<=15 THEN  GO TO 7
0
  80 DEF FN r(y,x)=ABS y^x-2*ABS
 y^x*((INT ((x+1)/2)-INT (x/2))*
(ABS y-y)/(2*ABS y))
  90 LET in=1e10: LET nuevo=in-9
9e8-7

 100 REM Dibujo de la pantalla
 110 RESTORE 1500
 130 READ x
 140 READ y,p$
 150 PRINT AT x,y;p$
 160 READ x
 170 IF x<>99 THEN  GO TO 140
 180 PLOT 0,136: DRAW 255,0
 190 PLOT 0,32: DRAW 255,0
 200 REM Pedir a(n)
 210 INPUT "a(n)= ";f$: PRINT AT
 1,6;f$
 220 DEF FN a(n)=VAL f$

 230 REM Limites
 240 IF FN a(1)>FN a(2) AND FN a
(2)<FN a(3) THEN  GO TO 300
 250 IF FN a(1)<FN a(2) AND FN a
(2)>FN a(3) THEN  GO TO 300
 260 PRINT AT 7,1;"limite a(n)=
"'" n-AB"
 270 LET li=FN a(in): LET pu=1
 280 LET x=7: LET y=13: LET pa=l
i
 290 GO SUB 2000: LET li=pa: GO
TO 470
 300 REM puntos de acumulacion
 310 LET pu=2: LET ps=FN a(in-99
e8-2)
 320 PRINT AT 7,1;"Punto sup.="'
" Punto inf.= "
 330 IF FN a(1)<FN a(3) AND FN a
(2)<FN a(4) THEN  GO TO 370
 340 IF FN a(1)>FN a(3) AND FN a
(2)>FN a(4) THEN  GO TO 370
 350 LET pi=FN a(in-99e8-1)-FN a
(in)+ps
 360 GO TO 380
 370 LET pi=FN a(in-99e8-1)+FN a
(in)+ps
 380 LET ps=FN a(in)
 390 IF ps<pi THEN  LET ps=pi: L
ET pi=FN a(in)
 400 LET x=7: LET y=12: LET pa=p
s
 410 GO SUB 2000: LET ps=pa
 430 LET x=8: LET y=12: LET pa=p
i
 440 GO SUB 2000: LET pi=pa

 500 REM Cotas
 510 IF pu=1 THEN  LET a1=FN a(1
): LET a2=a1: LET a=li: LET b=a:
 GO TO 560
 530 LET a1=FN a(1): LET a2=FN a
(2)
 540 LET a=ps: LET b=pi
 550 IF a1<a2 THEN  LET a2=a1: L
ET a1=FN a(2)
 560 IF a>a1 THEN  LET cs=a: LET
 s$="["
 570 IF a<=a1 THEN  LET cs=a1: L
ET s$="]"
 580 IF b<a2 THEN  LET ci=b: LET
 t$="]"
 590 IF b>=a2 THEN  LET ci=a2: L
ET t$="["
 600 IF ABS cs<=1e-8 THEN  LET c
s=0
 610 IF ABS ci<=1e-8 THEN  LET c
i=0
 620 IF cs>=in THEN  GO TO 620
 630 PRINT AT 12,14;s$;cs;",AB["
 640 IF ci<=-in THEN  GO TO 650
 650 PRINT AT 12,3;"]-AB,";ci;t$

 700 REM graficos inicializacion
 710 IF cs=ci THEN  LET px=128:
LET pr=50: LET ic=cs: GO TO 810
 720 IF cs<in AND ci>-in THEN  L
ET ic=(cs+ci)/2: LET px=128: GO
TO 790
 730 IF cs<in AND ci<=-in THEN
LET ic=cs: LET px=239: GO TO 770
 750 IF ci>-in THEN  LET ic=ci:
LET px=16: GO TO 770
 760 LET ic=FN a(1)+FN a(2): LET
 px=128
 770 LET pr=8
 780 GO TO 820
 790 IF cs-ci>28 THEN  GO TO 770
```




```
800 LET pr=224/(cs-ci)

810 REM enteros de la sucesion
820 LET x=5: LET y=140
830 GO SUB 2100
840 REM enteros de a(n)
850 LET x=18: LET y=36
860 GO SUB 2100

900 REM Representacion sucesion
910 LET n=1: LET er=0: LET di=1
920 LET co=INT ((FN a(n)-ic)*pr
)+px
930 IF co<0 OR co>254 THEN  GO
TO 980
940 PLOT co,143: DRAW 0,-5
950 IF ABS ((FN a(n)-FN a(n+2))
*pr)<2 THEN  LET di=di*SQR 3
960 IF di>100 THEN  GO TO 1000
970 LET n=n+di: GO TO 920
980 LET er=er+1
990 IF er<3 THEN  LET n=n+1: GO
TO 920

1000 REM creciente o decreciente
1010 LET ig=0: LET ma=0: LET me=
0
1020 LET n=1
1030 LET a1=FN a(n): LET a2=FN a
(n+1)
1040 IF a1=a2 THEN  LET ig=1: GO
 TO 1070
1050 IF a1>a2 THEN  LET ma=1: GO
 TO 1070
1060 IF a1<a2 THEN  LET me=1
1070 LET n=n+1
1080 IF n<5 THEN  GO TO 1030
1090 IF ig=1 THEN  LET c$="Monot
ona "
1100 IF ig=0 THEN  LET c$="Estri
ctamente "
1110 IF ma=1 AND me=0 THEN  LET
c$=c$+"Decreciente"
1120 IF ma=1 AND me=1 THEN  LET
c$=""
1130 IF ma=0 AND me=1 THEN  LET
c$=c$+"Creciente"
1140 PRINT AT 14,1;c$

1200 REM convergente, divergente
1210 IF pu=2 THEN  PRINT AT 15,1
;"Oscilante": GO TO 1250
1220 IF ABS li>=in THEN  PRINT A
T 15,1;"Divergente": GO TO 1250
1230 PRINT AT 15,1;"Convergente"

1300 REM Representacion a(n)
1310 LET co=0
1340 INPUT "n= ";n
1350 IF n=nuevo THEN  CLEAR : GO
 TO 80
1360 PRINT AT 20,1;"a";n;"=";FN
a(n);"          "
1370 PRINT AT 16,INT (co/8);" "
1380 LET co=INT ((FN a(n)-ic)*pr
)+px
1390 IF co<0 OR co>254 THEN  LET
 co=0: GO TO 1330
1400 PLOT co,40: DRAW 0,5
1410 GO TO 1330
1500 DATA 1,1,"a(n)=",10,4,"cota
",10,15,"cota"
1510 DATA 11,2,"inferior",11,13,
"superior"
1520 DATA 99
1530 DATA 0,0,24,38,65,70,56,0
1540 DATA 0,0,24,100,130,98,28,0

2000 IF pa>=in THEN  PRINT AT x,
y;"AB": RETURN
2010 IF pa<=-in THEN  PRINT AT x
,y;"-AB": RETURN
2020 IF ABS pa<=1e-8 THEN  LET p
a=0
2030 PRINT AT x,y;pa
2040 RETURN

2100 REM enteros
2110 LET n=1: LET di=1
2120 LET co=INT ((INT ic+n-ic)*p
r)+px
2130 IF co<0 OR co>254 THEN  GO
TO 2180

2140 PLOT co,y: DRAW 0,-3
2150 IF pr>17 OR (ic+n)/2=INT ((
ic+n)/2) THEN  PRINT AT x,INT (c
o/8);INT (ic+n)
2160 LET n=n+di
2170 GO TO 2120
2180 IF di=-1 THEN  RETURN
2190 LET di=-1
2200 LET n=0
2210 GO TO 2120
```



APARTADO DE CORREOS

Dirige tus cartas a:
Todospectrum
Bravo Murillo, 377, 5.º-A
28020 Madrid

ATRIBUTOS ILEGALES

En relación al programa «Editor de pantallas» del mes de octubre (n.º), tengo dudas que me gustaría que me resolvieran. En las líneas 60, 2170, 3050 y 4570 aparece una signo trás la sentencia PRINT que no encuentro en mi Spectru. Además en la línea 9000 encontramos INK 8, PAPER 8 y BRIGHT 8, cuando según mis datos estos comandos no pueden tomar estos valores.

Miguel A. López
Marbella (Málaga)

El símbolo que aparece tras PRINT en las líneas que comentas deberás sustituirlo por el de numeral, que podrás conseguir pulsando simultáneamente SIMBOL SHIFT y 3. El hecho de que aparezca un enigmático triángulo en su lugar es culpa de nuestra impresora, que, como ya hemos comentado repetidas veces, se empeña en escribirlo «a su modo».

En cuanto a INK, PAPER o BRIGHT 8, sí son comandos «legales» y lo que hacen es que cuando escribamos algo en la pantalla sean respetados los atributos (colores, etc) que hubiera en esta anteriormente.

GENS Y WAFADRIVE

Tengo el ensamblador GENS-3M y, debido a la agobiante lentitud del cassette, me he decidido a comprar en breve el Wafadrive. Es molesto tener que perder el tiempo realizando VERIFYcaciones cada vez que, al realizar una rutina, la tengo que salvar en cassette (teniendo en cuenta que salvarlo también lleva su tiempo), operación que hay que repetir tantas veces como veces se te «cuelga» la rutina al correrla, después de haber hecho las modificaciones pertinentes.

La pega está en que el GENS según creo (pues no tengo las instrucciones), no posee opción de salvado ni carga por Microdrive. Quisiera saber si es posible que me realizaran una rutina o una serie de modificaciones en el programa para subsanar dicho problema. Si fuera posible les agradecería que me dieran el nombre de algún ensamblador que sí posea esta opción.

Jorge Mejías
Palma de Mallorca

Lo primero que debemos decirte es que el hecho de que no tengas las instrucciones del programa que mencionas nos hace sospechar que tienes una versión «pirata» del mismo. En el caso de que esto fuera así no pensamos que fuera ético resolverte el problema para que pudieras disfrutar de un programa que no te pertenece. Pero, como no podemos estar seguros de sí realmente, es así o es que se te han perdido las instrucciones...

Las últimas versiones del GENS sí que admiten trabajar con microdrives. Para ello se utilizan los mismos comandos que para el cassette pero incluyendo el número de la unidad y «:»antes del nombredel fichero. Por ejemplo, para salvar en el microdrive 1 las líneas 1-200 de un programa llamado «PEPITO» haremos: P1,200,1:PEPITO.

El programa comprueba si el fichero que pretendemos salvar ya existe en ese cartucho, y, si es así, pide que confirmemos si queremos sobregrabar. El comando «H» sin argumentos hace un VERIFY del último fichero salvado.
tema operativo y el SUPERBASIC, a más versatilidad de uso de un lenguaje en un mismo número de Ks siempre corresponde una menor seguridad ante los posibles fallos del programador.

¿128 COMPATIBLE?

Hace unos meses salió al mercado el nuevo ordenador Sinclair, el Spectrum 128K, y aunque he leído el artículo publicado en TODOSPECTRUM explicando algunas de sus características, aún me quedan algunas dudas que espero me puedan resolver a través de esta revista: ¿es el BASIC del 128K similar al del antiguo Spectrum? ¿Puede emplearse un programa de Spectrum 48K en el 128K? Gracias por su ayuda.

Sixto M. Buendía
Almería

El BASIC del Spectrum 128K es muy similar al de los modelos anteriores, de hecho las 16K bajas de la

ROM del 128 son una copia exacta de la de aquellos. Las principales diferencias son: un nuevo editor de líneas, el hecho de que los comandos se tecleen letra a letra, su teclado numérico y la implementación de algunos comandos nuevos. Estos últimos son los que se utilizan para controlar los chips de sonido y comunicaciones, el uso de la memoria extra como disco RAM y el comando SPECTRUM, que convierte el 128K en un 48K normal y corriente. Cuando ejecutamos ese comando la compatibilidad con el viejo Spectrum es prácticamente total, por lo que no hay ningún problema en utilizar un programa original de éste.

LENGUAJE MAQUINA A DISCRECION

Me gusta mucho su revista, y les felicito por la serie de lenguaje máquina, pues me daba rabia ver un listado en ensamblador sin entenderlo (no es que ahora lo entienda, pero con el tiempo...). Lo único malo es que le asignan pocas páginas a esa sección, pues en el último número sólo fueron dos páginas; debería tener una extensión de 5 páginas como mínimo. Otra cosa a destacar y elogiar es la poca publicidad que presenta su revista.

En fin, les escribo para que me contesten a las siguientes cuestiones:

¿Me podrían indicar un libro de lenguaje máquina para novatos? La Ramtop indica el final de la memoria disponible para el Basic, pero... ¿Qué es? ¿Es acaso un número de dos bytes? Ordenadores como los MSX y al Amstrad poseen un Z-80, sin embargo, no se pueden confeccionar programas en C/M que sirvan tanto para el Spectrum como para los otros. ¿Por qué? ¿Son distintas instrucciones? (por favor, expliquen esto lo más claramente posible).

José M. Baleato
Santiago (La Coruña)

Nos alegramos de que guste nuestra revista. Por esto sí vale la pena dejarse el gorro mes a mes para que llegue a vuestras manos.

En cuanto a la serie de lenguaje máquina, como habrás visto, venimos dedicándole algo más que esas dos páginas del n.º 17. Esperamos que pronto puedas adentrarte en el desensamblador de esas rutinas que tanto te cuesta digerir.

Respondemos a tus preguntas:

Es difícil aconsejar un libro concreto de código máquina «para novatos», casi todo el material publicado en castellano adolece de un grave error: parte de un nivel demasiado elevado o lo hace de forma incompleta y mal estructurada. Por otra parte quien de una forma u otra ha alcanzado cierto nivel se encuentra con que ningún libro cubre totalmente los pequeños huecos que siempre quedan. La única alternativa consiste en ir «picando» aquí y allá, en libros y revistas, y dedicarle muchas horas. Sólo podemos recomendarte que ojees todo lo que caiga en tus manos y elijas el que más te guste.

Ramtop es una variable del sistema que indica al operativo cuál es la dirección de memoria más alta de las que pueden ser utilizadas para el almacenamiento del programa BASIC (+ variables, espacio de trabajo, etc.). Esta en realidad queda algo más abajo pues en esa dirección se encuentra normalmente la base de la pila de GOSUBs y el stack o pila de máquina, por lo que el sistema reserva siempre un cierto espacio para esto. Pero, ¿por qué poner límite a la RAM?

Aunque trabajemos en BASIC suele ocurrir que queramos reservar un espacio de memoria para dato o para alguna rutina de código. Si no pusieramos límite, el programa BASIC podría crecer hasta «tragarse» nuestros datos como si de un sandwich de jamón y queso se tratara. Podemos situar Ramtop donde queramos mediante el comando CLEAR, que debe ir seguido del nuevo valor que queramos asignarle. Cuando se inicializa la máquina se coloca Ramtop 168 bytes por debajo del límite máximo de la memoria, este espacio es utilizado para almacenar los caracteres gráficos definidos por el usuario. Efectivamente, Ramtop es un número que ocupa dos bytes, puedes calcular su valor haciendo PRINT PEEK 23730+256*PEEK 23731.

El hecho de que dos ordenadores usen el mismo microprocesador no es sinónimo de que puedan ser programados en lenguaje máquina de igual modo. Desde luego las instrucciones que utilicemos serán las mismas, pero la memoria estará distribuida de distinta forma, las llamadas a la ROM dependerán del operativo que utilice, y todos los dispositivos de entrada y salida estarán controlados por distintos puertos. A diferencia de los lenguajes de alto nivel, que con pocos cambios pueden ser utilizados en cualquier ordenador de la misma forma, el ensamblador exige al programador meterse hasta el cuello en la arquitectura de la máquina, «hurgando» en cada posición de memoria, en cada rutina de la ROM y en cada port para lograr sus objetivos.

La portada del número 17 la hizo «uno de nosotros», casualmente quien escribe estas líneas.

ERRATA DEL QL

Por favor, prevengan a sus lectores de este fallo de la ROM del QL (por lo menos en la versión MGE):

Sin nada en la memoria hace esto:

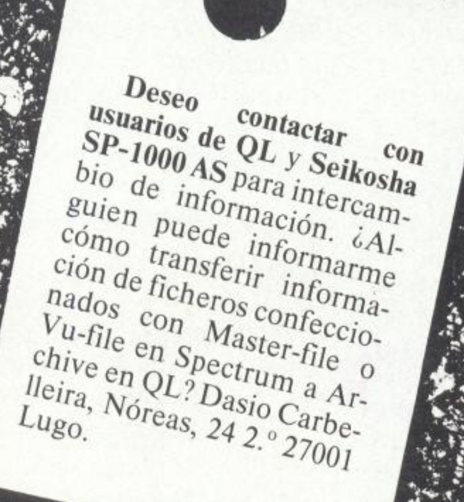
10 DEFine FuNction hola$(a)
20 END DEFine

Ahora... hola$(1)
¡Sorpresa!

Arturo Ramdírez
Majadahonda (Madrid)

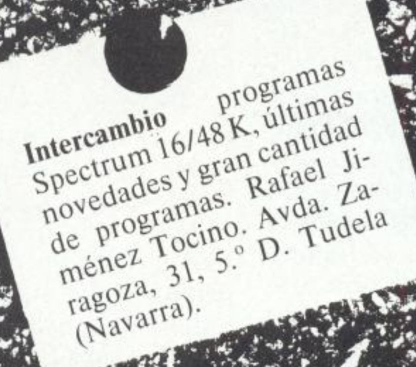
En efecto nuestro QL se «cuelga» o hace cosas muy raras. Se trata de una pequeña errata de la correspondiente rutina de la ROM, que no comprueba exhaustivamente las posibles «burradas» que pueda hacer el programador. Este tipo de detalles, que es más frecuente de lo que debiera en una máquina como el QL, se deben a la presión a que se debieron ver sometidos los programadores-diseñadores por parte de las altas esferas de Sinclair, ansiosas de sacar al mercado el nuevo producto. También influye en ello la filosofía usada a la hora de crear el sistema operativo y el SUPERBASIC, a más versatilidad de uso de un lenguaje en un mismo número de Ks siempre corresponde una menor seguridad ante los posibles fallos del programador.

EL CORCHO

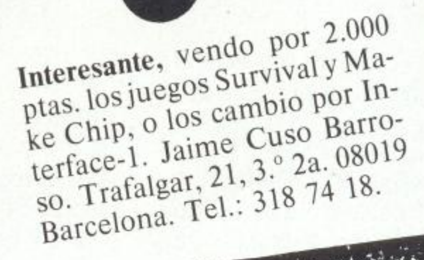

Deseo contactar con usuarios de QL y Seikosha SP-1000 AS para intercambio de información. ¿Alguien puede informarme cómo transferir información de ficheros confeccionados con Master-file o Vu-file en Spectrum a Archive en QL? Dasio Carballeira, Nóreas, 24 2.° 27001 Lugo.

Intercambio o vendo programas para el QL. Tengo compiladores, utilidades y juegos. Busco Cobol. Juan Carlos Ordóñez. Ferroviarios, 11. 28026 Madrid. Tel.: (91) 476 25 39.


Vendo por cambio de equipo Spectrum 48 K, manuales, etc. Con teclado profesional DK'Tronic, impresora Seikosha GP-50S, TV 12", interface joystick, joystick Quickshot II, cuadro de interruptores y toma de red con amplificador de sonido para Spectrum integrado, peana de soporte de televisión, cassette Sanyo periodista, cassette Satsonic, 25 números de Todospectrum, 25 de ZX, 60 de Microhobby, etc. Todo en perfecto estado por 85.000 ptas. Llamar al Tel.: (94) 458 01 17 de 22 a 23 h. Miguel.


Intercambio programas Spectrum 16/48 K, últimas novedades y gran cantidad de programas. Rafael Jiménez Tocino. Avda. Zaragoza, 31, 5.° D. Tudela (Navarra).

Quiero intercambiar instrucciones de programas para Spectrum. Si estáis interesados, enviar lista a: Mario Sáenz de Santamaría. Río Ebro, 27, 7.° C. Miranda de Ebro (Burgos).


Interesante, vendo por 2.000 ptas. los juegos Survival y Make Chip, o los cambio por Interface-1. Jaime Cuso Barroso. Trafalgar, 21, 3.° 2a. 08019 Barcelona. Tel.: 318 74 18.

Todo sobre AMSTRAD

1ª FERIA INFORMATICA

¡Ven a conocer el apasionante mundo de los ordenadores Amstrad!

Las más importantes empresas españolas y europeas del sector se dan cita en Madrid para presentar y ofrecer sus más recientes productos para **AMSTRAD.**

Programas de acción, juego, aventuras... Programas educativos, de utilidades, lenguajes... Programas de gestión y profesionales... Cientos de títulos inéditos...

Periféricos, ampliaciones de memoria, emuladores, tabletas gráficas, digitalizadores, impresoras, lápices ópticos, redes de comunicación, discos duros, sintetizadores de voz, correo electrónico, tratamiento de imágenes...

Las últimas novedades editoriales... Todas las revistas...

Una ocasión única para conocer de "primera mano" los increíbles ordenadores personales **AMSTRAD** y todo cuanto para ellos se produce en el mundo.



- Patrocinada y organizada por **AMSTRAD ESPAÑA**
- Horario continuo de **10:00 a 19:30**
- Entrada: **200 ptas.**
- Sorteo de Ordenadores **AMSTRAD** entre los visitantes.

23-24-25 MAYO

Palacio de Exposiciones y Congresos de Madrid

P.º Castellana, 99. 28046 MADRID